INTERNET-DRAFT
Category: Experimental
Expires: Nov 1, 2004

Hadmut Danisch

May 2004

The RMX DNS RR and method for lightweight SMTP sender authorization
draft-danisch-dns-rr-smtp-04.txt

Status of this Memo

Abstract

This memo introduces a new authorization scheme for SMTP e-mail transport. It is designed to be a simple and robust protection against e-mail fraud, spam, and worms. It is based solely on organisational security mechanisms and does not require but still allow use of cryptography. This memo also focuses on security and privacy problems and requirements in context of spam defense.

This document is part of the LMAP work of the Anti-Spam Research Group (ASRG) of the IRTF.

Table of Contents

1. Problem and threat description

1.1. Mail sender forgery

The amount of e-mails with forged sender addresses has dramatically increased. As a consequence, damages and annoyances caused by such e-mails increased as well. In the majority of examined e-mails the domain name of the envelope sender address was forged, and the e-mail was sent from an IP address which does not belong to a network used by the actual owner of the domain.

1.1.1. Definition of sender forgery

As discussions, comments to prior drafts of this RFC, and different approaches to stop forgery showed, different perceptions of "mail forgery" exist. For example, there are mechanisms to verify e-mail addresses for mailing lists, web servers, or to stop spam, which do send a message with a random number to the given address and expect the user to send a reply. Here, someone is considered to be allowed to use a particular e-mail address, if and only if he is able to receive messages sent to this address, and is able to reply to such a message. While this definition appears to be quite plausible and natural, it can't be used for a simple technical solution. Sending back a challenge and expecting a reply is simply too much overhead and time delay, and not every authorized sender is able and willing to reply (e.g. because he went offline or is not a human).

Within the scope of this memo, sender forgery means that the initiator of an e-mail transfer (which is the original sender in contrast to relays) uses a sender address which he was not authorized to use. Being authorized to use an address means that the owner (administrator) of the internet domain has given permission, i.e. agrees with the use of the address by that particular sender. This memo will cover both the permission of the full e-mail address and the domain part only for simplicity.

Within context of Internet and SMTP, the sender address usually occurs twice, once as the envelope sender address in SMTP [1], and once as the address given in the mail header [2]. While the following considerations apply to both addresses in principle, it is important to stress that both addresses have distinct semantics

and are not necessarily the same. The envelope address identifies the initiator of the transport, while the header identifies the author of the message content. Since this memo deals with the message transport only and completely ignores the message content, the method should naturally be applied to the envelope sender address. However, this is currently under discussion in the ASRG

and the IETF working groups.

1.1.2. Spam

A common and well known problem is the dramatic increase of unsolicited e-mail, commonly called "spam". Again, the majority of examined e-mails had forged sender addresses. The abused domains were mainly those of common webmailers as hotmail or yahoo, or well-known companies.

Unfortunately, there is no accurate definition of spam available yet, and neither are there concise technical criterions to filter or block spam with technical mechanisms. There are efforts to design content based filters, but these filters are expensive in calculation time (and sometimes money), and they do not reliably produce predictable results. They usually give false positives and/or require user interaction. Content filters in general suffer from a design problem described later in this memo. Therefore, this proposal does not use the content based approach to block spam.

As analysis of spam messages showed, most of spam messages were sent with forged envelope sender addresses. This has mainly three reasons. The first reason is, that spam senders usually do not want to be contacted by e-mail. The second reason is, that they do not want to be blacklisted easily. The third reason is, that spam is or is going to be unlawful in many countries, and the sender does not want to reveal his identity. Therefore, spam is considered to be a special case of sender forgery throughout this memo.

1.1.3. E-Mail Worms

Another example of sender forgery is the reproduction of e-mail worms. Most worms use random sender addresses, e.g. the addresses found in mailboxes on the infected system. In most cases analyzed by the author, the e-mails sent by the reproduction process can also be categorized as forged, since the infected system would under normal circumstances not be authorized to send e-mails with such e-mail addresses. So forgery does not require a malicious human to be directly involved. This memo covers any kind of e-mail sender address forgery, included those generated by malicious

software.

1.1.4. E-Mail spoofing and fraud

Forging e-mail sender addresses for fraud or other kinds of deception ("human engineering") has also dramatically increased.

There are many known cases where single or mass e-mails were sent with false sender addresses, pretending to come from service providers, software manufacturers etc., and asking the receiver to install any software or patches, or to reply with any confidential information. The Internet is increasingly becoming a scene of crime, and so are it's services, including e-mail. It is obvious that crime based on e-mail is eased by the fact that SMTP allows arbitrary sender address spoofing.

1.2. Indirect damage caused by forgery

As observed by the author, mass mails and worms with forged sender addresses can cause a severe damage for the real owner of the abused sender addresses. If a sender A is sending an e-mail to the receiver B, pretending to be C by using a sender address of C's domain, then C has currently no chance to prevent this, since C's machines and software are not involved in any way in the delivery process between A and B. B will nevertheless send any error messages (virus/spam alert, "no such user", etc.) to C, erroneously assuming that the message was sent by C. The author found several cases where this flood of error messages caused a severe denial of service or a dramatic increase of costs, e.g. when C was downloading the e-mail through expensive or low bandwidth connections (e.g. modem or mobile phones), or where disk space was limited. The author examined mass mailings, where several tens or hundreds of thousands of messages were sent to recipients around the world, where these messages caused only annoyance. But since several thousands of these recipient addresses were invalid or didn't accept the message, the owner of the DNS domain which was abused by the spammer to forge sender addresses was flooded for several months with thousands of error messages, jamming the e-mail system and causing severe costs and damages.

As a consequence, when A sends a message to B, pretending to be C, there must be any mechanism to allow C to inform B about the fact, that A is not authorized to use C as a sender address. This is what this memo is about.

1.3. Technical problem analysis

Why does e-mail forgery actually exist? Because of the lack of the Simple Mail Transfer Protocol SMTP[1] to provide any kind of sender authentication, authorization, or verification. SMTP was designed

at a time where security was not an issue. Efforts have been made to block forged e-mails by requiring the domain part of the sender address to be resolvable. This method provides protection from e-mails with non-existing sender domains, and indeed, for some time it blocked most spam e-mails. However, since attackers and spam

senders began to abuse existing domain names, this method was rendered ineffective.

1.4. Shortcomings of cryptographical approaches

At a first glance, the problem of sender address forgery might appear to be solvable with cryptographical methods such as challenge response authentications or digital signatures. A deeper analysis shows that only a small, closed user group could be covered with cryptographical methods. Any method used to stop spam forgery must be suitable to detect forgery not only for a small number of particular addresses, but for all addresses on the world. An attacker does not need to know the secrets belonging to a particular address. For him it is sufficient to be able to forge any address and thus to know any secret key. Since there are several hundreds of millions of users, there will always be a large amount of compromised keys, thus spoiling any common cryptographic method. Furthermore, cryptography has proven to be far too complicated and error prone to be commonly administered and reliably implemented. Many e-mail and DNS administrators do not have the knowledge required to deal with cryptographic mechanisms. The most important requirement for a world wide applicable spam protection is simplicity. Many legislations do not allow the general deployment of cryptography and a directory service with public keys. For these reasons, cryptography is applicable only to a small and closed group of users, but not to all participants of the e-mail service.

After all, after more than 20 years of Public Key Cryptography, there is still no common Public Key Infrastructure, there is still not enough adequate crypto software available, neither hardware devices, and existing crypto software is far from being robust and free of severe bugs. Cryptography cannot be expected to solve the spam problem in the foreseeable future.

2. A DNS based sender address verification

2.1. Overview

To gain an improvement in e-mail authenticity while keeping as much SMTP compatibility as possible, a method is suggested which doesn't change SMTP at all.

The idea is to store the information about how to verify who is authorized to transmit e-mails through SMTP with a particular sender address (either full address or - for simplicity - only the domain part of the address) in a directory service. The internet's directory service is currently DNS. To be precise, the verification consists of two steps, the classical pair of authentication and authorization:

The first step is the authentication. While several methods are possible to perform authentication (see below), the most important and robust method is the verification of the sender's IP address. This is done implicitely by TCP/IP and the TCP sequence number. The authenticated identity is the IP address. It has to be stressed that this TCP/IP "authentication" is a weak authentication and vulnerable to several attacks. It is nevertheless sufficient for this purpose, especially for blocking spam. It doesn't take any implementation and it doesn't cost: It is already there, it is a functionality of TCP/IP. An incoming SMTP connection based on TCP/IP already carries the sender's IP address without any modification of SMTP. See below (section Entry types) for more details about authentication methods.

The second step is the authorization. It is based on the identity given by the previous authentication step, e.g. the IP address of the originator of the incoming SMTP connection, and on the envelope sender address. The mechanism proposed in this memo answers the question "Is that particular sender (IP address,...) allowed to send with that sender address" by querying and processing authorization records stored in a directory service, which is DNS.

When the sender has issued the "MAIL FROM:" SMTP command, the receiving mail transfer agent (MTA) can - and modern MTAs do - perform some authorization checks, e.g. run a local rulebase or

check whether the sender domain is resolvable.

The suggested method is to let the DNS server for the sender domain provide informations about who - this means for example which IP address - is authorized to use an address or a domain as a part of it. After receiving the "MAIL FROM:" SMTP command, the receiving

MTA can verify, whether e. g. the IP address of the sending MTA is authorized to send mails with this domain name. Therefore, a list of entries with authorized IP addresses or other descriptions is provided by the authoritative DNS server of that domain. The entry types are described in the subsequent chapters. Some of these entry types are

- An IPv4 or IPv6 network address and mask
- A fully qualified domain name referring to an A record
- A fully qualified domain name referring to an APL record

RMX records of these types would look like this:

```
somedomain.de.      IN RMX ipv4:10.0.0.0/8
rmxtest.de.         IN RMX host:relay.anyprovider.com
danisch.de.         IN RMX apl:relays.rackland.de
relays.rackland.de. IN APL 1:213.133.101.23/32 1:1.2.3.0/24
```

where the machine with the example address 213.133.101.23 and the machines in the example subnet 1.2.3.0/24 are the only machines allowed to send e-mails with an envelope sender address of domain danisch.de. Since the APL records do not necessarily belong to the same domain or zone table as the RMX records, this easily allows to refer to APL records defined by someone else, e.g. the internet access or server hosting provider, thus reducing administrative overhead to a minimum. In the example given above, the domain danisch.de and several other domains are hosted by the service provider Rackland. So if the relay structure of Rackland is modified, only the zone of rackland.de needs to be updated. The domain owners don't need to care about such details.

2.2. Envelope vs. header sender address

Questions were raised why the proposed mechanism is based on the envelope sender address, and not on the sender address given in the message header. Technically, both can be used. Actually, it makes sense to use the envelope address.

In common, the header sender address identifies the author of the content, while the envelope sender tells who caused the transmission. The approach proposed in this memo is transmission

based, not content based. We can not authorize the author of a message if we don't have contact with him, if the message does not already contain a signature. In contrast, the sending MTA is linked to an IP address which can be used for authentication. This mechanism might not be very strong, but it is available and sufficient to solve today's e-mail security problems.

Some people argued that it is the header address and not the sender address, which is displayed in common mail readers (MUAs), and where the receiver believes the mail to come from. That's true, but it doesn't help. There are many cases where the header sender differs from the envelope sender for good reasons (see below in the consequences chapter for the discussion about relaying). Relaying, mailing lists etc. require to replace the sender address used for RMX. If this were the header address, the message header would have to be modified. This is undesirable.

2.3. Domain part vs. full sender address

Early draft versions of this memo were limited to the domain part of the sender address. The first reason is that it is common and MX-like, to lookup only the domain part of an e-mail address in DNS. The second reason is, that it was left to the private business of the domain administration to handle details of user verification. The idea was that the domain administration takes care to verify the left part of an e-mail address with an arbitrary method of their individual taste. RMX was originally designed to ignore the left part of the address and to expect the domain administration to take over responsibility for enforcing their policy. If, e.g., a spam message arrived and passed the RMX mechanism, it is known to be authorized by the domain administration and they can be blamed, no matter what is on the left side of the sender address - it's their private problem what happens on the left side of the @. By far the most of the comments to prior draft versions of this memo agreed with that. A few comments asked for a finer granularity.

And indeed, there is no technical reason against a finer granularity. All it takes is a mapping from a given envelope sender address to a DNS name, and the RMX lookup for that particular e-mail address could be done instead of a lookup for the domain part only. However, to my knowledge, most domain administrators would not like to provide an RMX entry for every single e-mail address. In many cases, this would also overload DNS servers.

It is to be discussed how to cover both views. One method could be to query the full address, and if no RMX records were found to query the domain part only. A different approach would be to query the domain part only, and if it's RMX record contains a special entry, then a new query for the full address is triggered. A third

way would be to always query the full address and to leave the problem to the wildcard mechanism of DNS.

A completely different approach to allow authorization with full

address and even much finer granularity is the RMX++ proposal mentioned in the future development section below.

3. Mapping of E-Mail addresses to DNS names

To perform the RMX query, a mapping is needed from E-Mail addresses to DNS fully qualified domain names. In other words: A function is needed which tells for every incoming e-mail where in DNS to look for authorization records.

This chapter is only a rough outline. Details are currently under discussion in the ASRG and IETF working groups.

3.1. Domain part only

Mapping of the domain part is trivial, since the domain part of an e-mail address itself is a valid DNS name and does not need translation. It might be nevertheless desirable to distinguish the RMX entries from other entries, depending of the encoding of the records. If the RMX entries are encoded in TXT record types, they might collide with other uses of TXT records. It might be necessary to prepend the domain part with a special prefix, e.g. _rmx. So the e-mail address some.user@example.com could be mapped to example.com or _rmx.example.com.

3.2. Full address

Mapping a full address is slightly more difficult. The @ symbol must be unambiguously translated, and therefore can not be simply translated into a dot. The e-mail addresses some.user@example.com and some@user.example.com must have different mappings. Therefore, the @ symbol could be translated into _rmx, implicitely assuming that this is not an allowed domain name component of normal domain names. Then the rightmost _rmx in the mapped DNS name always corresponds to the @ symbol. some.user@example.com would be translated into some.user._rmx.example.com and can be covered by a wildcard entry like *._rmx.example.com.

Character encoding and character sets are still to be discussed.

3.3. Empty address

Unfortunately, SMTP allows empty envelope sender addresses to be

used for error messages. Empty sender addresses can therefore not be prohibited. As observed, a significant amount of spam was sent with such an empty sender address. To solve this problem, the host name given in the HELO or EHLO command could be used instead to lookup the RMX records. This makes sense, since such messages were generated by the machine, not a human.

4. Mandatory entry types and their syntax

The entry types described in this section MUST be supported by all implementations of this memo.

4.1. Overall structure

Similar to APL, an RMX record is just a concatenation of zero or more RMX entries. The entries within one record form an ordered rule base as commonly usual in packet filtes and firewall rulesets, i. e. they are processed one ofter another until the first entry matches. This entry determines the result of the query. Once a matching entry is found, the RMX processing is finished.

For any domain name there should not exist more than a single RMX record. Due to the structure of DNS, it is nevertheless possible to have more than a single RMX record. Multiple RMX records are treated as a single record consisting of the concatenation of all records. While the entries in a record are ordered, the records are not ordered and may be processed in arbitrary order. If the order of the entries matters, it is the zone maintainer's responsibility to keep those entries in a single record. For example, there are negative entries, which exclude IP addresses from authorization. It is important that these entries are processed before positive entries giving permission to a wider address range. Since order is guaranteed only within a record, corresponding negative and positive entries must be put in the same record.

An RMX record may consist of one or more entries, where the entries are separated by whitespace. An entry must not contain white space. Each entry consists of an optional exclamation sign, a tag, a colon, and the entry data:

```
[!] TAG : ENTRY-SPECIFIC-DATA
```

If the entry starts with an exclamation sign, the entry is negated. See the entry type description below for details.

The TAG is the mnemonic type identifier or the decimal number of

the entry. The TAG is case-insensitive. It is immediately followed by a colon.

The syntax and semantics of ENTRY-SPECIFIC-DATA depends of the the entry type. See description below.

Example:

```
danisch.de.  IN RMX apl:relays.rackland.de !ipv4:1.2.3.5
                    ipv4:1.2.3.0/24
```

4.2. Unused

This is a primitive entry which just says that this sender address will never be used as a sender address under any circumstances. Example:

```
testdomain.danisch.de    IN RMX unused:
```

4.3. IPv4 and IPv6 address ranges

These entry types contain a bit sequence representing a CIDR address part. If that bit sequence matches the given IP address, authorization is granted or denied, depending on the negation flag.

The entry is prepended with the tag "IPv4" or "IPv6". The colon is followed with an IPv4 or IPv6 address in standard notation, optionally followed by a slash and a mask length. If the negation flag is set, then the given address range is excluded. Examples:

```
danisch.de   IN RMX ipv4:213.133.101.23 ipv6:fe00::0
             IN RMX ipv4:10.0.0.0/8     ipv6:fec0::0/16
             IN RMX !ipv4:1.2.3.4
```

(Please note that it does not make much sense to use RFC1918-Addresses in RMX records, this is just to give a syntax example.)

4.4. DNS Hostnames and Dynamic IP addresses

This entry type simply contains a regular DNS name, which is to be resolved as a host name (fetch the A record or IPv6 equivalent). If the given IP address matches the result, authorization is granted or denied, depending on the negation flag.

The entry is prepended with the tag "host", followed by a colon and

the hostname. Examples:

```
danisch.de  IN RMX host:relay.provider.de
            IN RMX !host:badmachine.domain.de apl:relays.domain.de
```

Several people argued against RMX that it would break their existing installation which delivers e-mail from dynamically assigned IP addresses, because their IP providers didn't assign a static address, or because they are road warriors, plugging their notebook in any hotel room on the world.

RMX provides a simple solution: If such a machine has a dynamically updated DNS entry (e.g. DynDNS), all it takes is an RMX entry of the hostname type pointing to this dynamic DNS entry.

The cleaner solution would be to deliver mail the same way as it is received: If downloaded by POP from a central relay with a static address, where the MX points to, then it would be a good idea to deliver e-mail the same way in reverse direction. Unfortunately, plain POP does not support uploading yet.

4.5. APL Reference

This entry type simply contains a regular DNS name, which is to be resolved as an APL record [3] index (fetch the APL record). If the given IP address positively matches the APL, authorization is granted. Details of the semantic (espially when the negation bit is set) are still to be defined. It is still to be defined how to treat unresolvable entries.

The entry is prepended with the tag "host", followed by a colon and the hostname. Example:

```
danisch.de     IN RMX apl:relays.rackland.de
```

4.6. Domain Member

In many cases it is desirable to cover all hosts of a given domain with an RMX record without the need to duplicate the list of these hosts. This entry type does it (thanks to Eric A. Hall for pointing out this entry type). It contains a regular DNS name.

If this entry type is given, a reverse DNS query for the IP address of the sending MTA is performed to find its official fully qualified domain name. To prevent spoofing, this domain name is accepted only if a subsequent address query to the given domain name points to exactly the IP address of the sending MTA (the usual procedure to verify PTR records).

The entry matches if the fully qualified domain name of the sending MTA ends in the given domain. The negation flag works as usual.

The tag for this entry type is "domain". After the colon the domain name is given, but might be empty, thus pointing to itself. Example:

```
somedomain.org  IN RMX domain:somedomain.org domain:provider.com
```

would authorize all machines which's hostname can be verified

through an PTR and A query, and which ends in "somedomain.org" or "provider.com".

With such an entry, large companies with different networks can easily be covered with just a single and simple RMX entry. Obviously, it requires proper PTR records.

As a special shortcut, the DNS name may be empty. In this case the domain name of the zone itself is taken. Thus, with a very simple entry of the type

```
somecompany.com  IN RMX domain:
```

a company could authorize all machines which's IP addresses map to DNS names end in somecompany.com, which applies in the majority of companies.

Thus, a simple entry of the form

```
@ IN RMX domain:
```

would be a good starting point for company networks and would in most cases allow easy and simple RMX configuration if the network can't be described with a simple network mask.

4.7. Full Address Query

As described above, RMX records will in most cases apply to the domain part of the sender address. In special cases it might be desirable to query the RMX record for a particular address. An RMX entry of the Full Address Query type may occur in a domain RMX record only. It signals that the RMX record for the full address is to be fetched and processed.

This entry type does not take arguments. The negation flag is not supported. The tag is "full".

If such a full address query is to be performed, the mail address must be mapped to a valid and non-ambiguos DNS name. This mapping is still to be defined. It is not sufficient to simply replace the @ with a dot, because of case sensitivity, character sets, etc. The e-mail addresses

```
john.doe@example.org
John.Doe@example.org
john@doe.example.org
```

must all be mapped to different DNS entries. A better approach is RMX++ (see below).

4.8. MX reference

This entry type has no parameters. It means that all those machines are authorized, which are pointed to by an MX record.

Example:

danisch.de. IN RMX MX:

would simply allow all machines receiving mails for danisch.de (i.e. the MX machines) to deliver as well.

5. Optional and experimental entry types

The following subsections roughly describe further experimental entry types. These methods are just considerations about what to include in RMX and what to not include. The main purpose of this section is to start a discussion about such entry types.

The disadvantage of the following methods is that they violate the basic idea of RMX, i. e. to be simple, robust, easy to implement and easy to administer. The author does not believe that it is a good idea or even feasible to implement cryptography for a world wide e-mail transfer network. Keep in mind that cryptographic keys can be copied. Even if only around 0.01% of the cryptographic keys are stolen, this still compromises and spoils RMX. Cryptography is simply the wrong tool for the problem RMX is intended to solve. It is nevertheless to be discussed.

5.1. TLS fingerprint

The sender is considered to be authorized if the message was transmitted through SMTP and TLS[4], and the sender used a certificate matching the fingerprint given in the RMX record.

5.2. TLS and LDAP

The receiver could perform an LDAP query for the sender address (through the LDAP SRV record or given in the RMX record), fetch the X.509 certificate for the sender. The sender is considered to be authorized when the message was transmitted through SMTP and TLS using this certificate.

5.3. PGP or S/MIME signature

It would be possible to accept a message only if it was signed with PGP or S/MIME with a key which's fingerprint is given in the RMX record or to be fetched from LDAP or any PGP database. This is just for discussion, since it violates the idea of RMX to focus on the transport, not on the content. It would also allow replay attacks and not cover the envelope sender address or message header.

5.4. Transparent Challenge/Response

It would also be possible to implement a challenge-response mechanism without modifying the syntax of SMTP. For example, the receiving MTA could issue a challenge with it's very first greeting message, the sending MTA could include the response in the HELO or EHLO parameter and when the receiving MTA later learns the sender

envelope address, it could verify the response based on entries in the RMX record.

5.5. SASL Challenge/Response

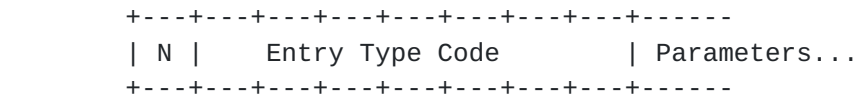
Modern SMTP implementations already include a SASL[5] mechanism, which easily allows to plugin new authentication mechanisms. While common SASL mechanisms require to use a previously shared password, a new mechanism could perform a challenge response authentication as a SASL method.

6. Encoding

6.1. RMX Records

6.1.1. Overall structure

Each entry starts with an octet containing the entry type and the negation flag:



N — If this bit (MSB) is set, an IP address matching this entry is not authorized, but explicitely rejected. See entry type descriptions for details.

Entry Type — A 7bit number simply determining the entry type.

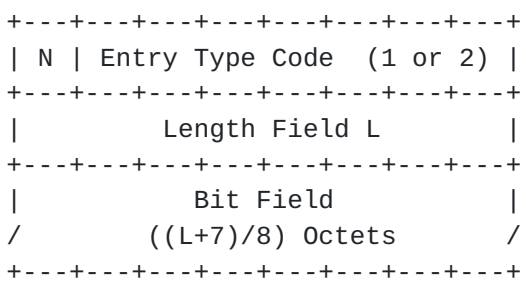
Currently, entries do not have an explicit length field, the entry length is determined implicitely by the entry type. Applications are required to abort if an unknown entry type is found, instead of skipping unknown entries.

6.1.2. Record encoding

A RMX record is simply a concatenation of RMX entries.

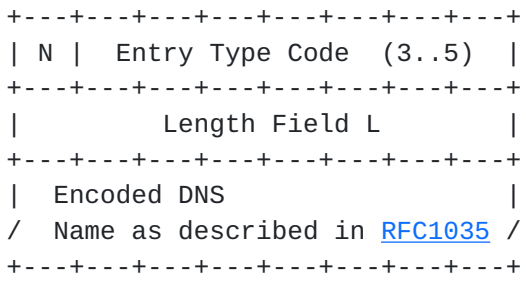
6.1.3. Encoding of IPv4 and IPv6 address ranges

After the entry type tag as described above, one octet follows giving the length L of the bit sequence. Then a sequence of exactly as many octets follows as needed to carry L bits of information (= trunc((L+7)/8)).


| N | Entry Type Code (1 or 2) |
| Length Field L |
| Bit Field |
/ ((L+7)/8) Octets /

6.1.4. Encoding of DNS

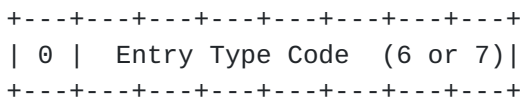
After the entry type tag immediately follows a DNS encoded[6] domain name.



In contrast to earlier draft versions of this memo, the DNS name cannot be compressed, since this would cause decompression errors when a DNS server which does not know this particular RR type is part of the query chain.

6.1.5. Encoding of unused and full address query

These entries do not contain parameters and does not allow the negation flag. So the encoding is quite simple:



6.1.6. Additional Records

In order to avoid the need of a second query to resolve the given host name, a DNS server should enclose the A record for that domain name in the additional section of the additional section of the DNS reply, if the server happens to be authoritative.

In order to avoid the need of a second query to resolve the given host name, a DNS server should enclose the APL record for that domain name in the additional section of the additional section of

the DNS reply, if the server happens to be authoritative.

6.2. Alternative encoding as TXT records

The main objection against the prior versions of this draft was that it requires a new RR entry type and upgrading all DNS servers.

Therefore and alternative encoding is proposed. Instead of using a new RR type, the TXT record type is used to contain the RMX record. The records would simply look as described in the entry type chapters above, e.g.

```
_rmx.danisch.de.   IN TXT "apl:relays.rackland.de"
```

To allow smooth introduction of RMX without the need to immediately upgrade all DNS servers, all clients (which have to be newly installed anyway) MUST support both the TXT and the RMX records. A client has to perform an ANY or a TXT and a RMX query. Servers/zone tables may currently use TXT entries but SHOULD use RMX entries in future.

7. Message Headers

An RMX query must be followed by any kind of action depending on the RMX result. One action might be to reject the message. Another action might be to add a header line to the message body, thus allowing MUAs and delivery programs to filter or sort messages.

In future, the RMX result might be melted into the Received: header line [2].

The details of such entries are to be discussed. As a proposal the following form is suggested:

X-RMX: RESULT addr ADDRESS by HOST on DATE mechanism MECHANISM

where

RESULT is one of "Granted", "Denied", "NotInRMX", "NoRMX", "TempFail", "BadData", "Trusted".

ADDRESS is the IP address of the sending machine

HOST is the name of the machine performing the RMX query.

DATE is the date of the query.

MECHANISM is the RMX method used to authorize the sender.

8. SMTP error messages

If a message is rejected because of RMX records, an error message should be issued which explains the details. It is to be discussed whether new SMTP error codes are to be defined. Error messages should be verbose to make debugging of configuration errors easy.

9. Message relaying and forwarding

9.1. Problem description

Message forwarding and relaying means that an MTA which received an e-mail by SMTP does not deliver it locally, but resends the message - usually unchanged except for an additional Received header line and maybe the recipient's address rewritten - to another SMTP MTA. Message forwarding is an essential functionality of e-mail transport services, for example:

- Message transport from outer MX relay to the intranet
- Message forwarding and Cc-ing by .forward or .procmail-alike mechanisms
- Mailing list processing
- Message reception by mail relays with low MX priority, usually provided by third parties as a stand-by service in case of relay failure or maintenance
- "Forwarding" and "Bouncing" as a MUA functionality

In all of these cases a message is sent by SMTP from a host which is not covered by the original sender domain's RMX records. While the RMX records would forbid accepting this message, it still must be accepted. The following subsections explain how to cope with relaying.

9.2. Trusted relaying/forwarding

In some cases the receiving MTA trusts the sending MTA to not fake messages and to already have checked the RMX records at message reception. As a typical example, a company might have an outer mail relay which receives messages from the Internet and checks the RMX records. This relay then forwards the messages to the several department's mail servers. It does not make sense for these department mail servers to check the RMX records, because the RMX records have already been checked and because they would always reject messages, since the relay is not covered by the originator's RMX records. In this case there is a trust relationship between the department relays and the outer relay. So RMX checking is turned off for trusted relays. In this example, the department relays would not check messages from the outer relay (but for intranet security, they could still check RMX records of the other

departments sub-domains to avoid internal forgery between departments).

Another common example are the low-priority MX relays, which receive and cache e-mails when the high-priority relays are down. In this case, the high-priority relay would trust the low-priority

relay to have verified the sender authorization and would not perform another RMX verification (which would obviously fail).

When a relay forwards a message to a trusting machine, the envelope sender address should remain unchanged.

9.3. Untrusted relaying/forwarding

If the receiving MTA does not trust the forwarding MTA, then there is no chance to leave the sender envelope address unchanged. At a first glance this might appear impracticable, but this is absolutely necessary. If an untrusted MTA could claim to have forwarded a message from a foreign sender address, it could have forged the message as well. Spammers and forgers would just have to act as such a relay.

Therefore, it is required that, when performing untrusted forwarding, the envelope sender address has to be replaced by the sender address of someone responsible for the relaying mechanism, e.g. the owner of the mailing list or the mail address of the user who's forwarding mechanism caused the transmission. It is important to stress that untrusted relaying/forwarding means taking over responsibility for the message. It is the idea of RMX records to tie responsibility to message transmission. Untrusted relaying without replacing the sender address would mean to transmit without taking responsibility.

The disadvantage is that the original sender address is lost. Therefore, whenever a sender address replacement happens, the Received-Line must contain the old address. Many of today's MTAs already insert the envelope recipient address, but not the sender address into the Received header line. It seems reasonable to require every Received line to include both the sender and recipient address of the incoming SMTP connection.

10. Further development and improvements of RMX

This RFC is intended to be close to the earlier RMX drafts for historical reasons. Therefore, the further development and improvements are not made "in place" but described in this new chapter.

10.1. Separate RMX records for address types

In earlier draft versions of this memo there was only one RMX record covering all possible identity classes provided by the authentication step, e.g. IPv4 and IPv6 addresses were described by the same RMX record.

This does not make sense and unnecessarily inflates the RMX records. Since the receiving MTA knows the identity class at query time, separate RMX records can be provided for each supported identity class, e.g. different RMX records for IPv4 and IPv6 addresses.

So RMX records could look like

```
_ipv4._rmx.danisch.de   IN RMX ipv4:213.133.101.23
_ipv6._rmx.danisch.de   IN RMX ipv6:fec0::0/16
```

10.2. SCAF - Simple Caller Authorization Framework

Fraud, spam, worms, spoofing are not limited to SMTP only. Other internet protocols like news transfer, chat and instant messaging, and even non-internet protocols can be protected against spoofing with RMX. It could also become a simple, password-less caller identification mechanism for protocols like HTTP or FTP. For example, a web browser could provide a user address similar to an e-mail address as a HTTP [7] cookie, in a new request header entry type, or as a password-less HTTP authentication header.

Imagine there is a vendor's web server providing web pages available for employees of a particular company only (e.g. software upgrades for a customer). The customer could be required to provide an RMX-like record describing it's network which allows the

web server to limit access based on the given user address and the RMX record for this application. (It can be understood as a kind of external firewall rule. Even firewalls and proxies could support it.)

To distinguish records for different applications, the records must be stored at different locations in DNS, e.g.

```
_ipv4._smtp._rmx.danisch.de   IN RMX ipv4:213.133.101.23
_ipv4._http._rmx.danisch.de   IN RMX host:homeoffice.danisch.de
```

would restrict access to web servers with "hadmut@danisch.de" to the IP address pointed to by the A record of homeoffice.danisch.de (which can be a dynamically assigned address).

10.3. RMX++

RMX has restrictions and might not be applicable or desirable in all cases due to it's inflexible record types, it's nature to reveal the domain's network structure, and limitations of DNS. The domain owner would always be limited to those entry types commonly defined.

To overcome these restrictions, the successor RMX++ significantly differs from RMX. With RMX++, the query is split into two distinct steps. In a first step, the receiving MTA takes the domain name of the sender address to query DNS as with RMX. But instead of an RMX record, the MTA queries an A record, an SRV record [8], and a TXT record.

The A or SRV records are then used to locate a server for the second step. The TXT record contains a URL pattern. A default pattern is used in absence of the TXT record. The pattern contains macros which are to be expanded, e.g. substituted with the server address, the mail sender address, the calling MTA's IP address, message ID, content type etc. The MTA then fetches the RMX record found at that URL. The preferred protocol type is HTTP or HTTPS, but other protocols could be used as well. Even DNS and LDAP queries can be described in URLs.

In general, there are three types of RMX records: In the first case, the RMX record is a static one and stored as a file on the web server. In the second case, it is dynamically generated by the web server (e.g. through a CGI program). In the third case, the MTA passes required information to the server and the server replies with "Allow" or "Reject". The receiving MTA does query and handle all three types the very same way.

This method has several advantages:

- No need for a new DNS RR type. DNS servers don't need to be upgraded.

- Easier to administer: Today, every domain owner is able to put a file on a web site. Helper programs run on the web server will help to generate the RMX records with

easy and foolproof user interfaces. In contrast, DNS zone tables are more difficult and not as easy available for modification for everyone at any time.

- There is no size limit for the record as with DNS. The RMX record does not need to be split into several DNS records and stitched together. The encoding does not need to be as tight as described above, and can be plain text, ASN.1, XML, etc.

- Simple encryption with HTTPS

- Support of full sender address verification (not just the domain part) is trivial.

- Caching with HTTP proxies, caching control and expiry with HTTP headers

- The RMX record can be generated dynamically on request. Large sites where thousands of users log in and out all the time (e.g. large mail service providers) can provide "fresh" records for every request without the need to update their zone table every second.

  Dynamic RMX records can be easily generated with CGI applications, a well known and robust mechanism.

- RMX records can be smaller because they need to cover only the query sent to the server and don't need to describe the full network structure which might consist of thousands of computers.

- RMX verification can be moved from the receiving MTA to the domain owner's server, because the MTA can pass all required data such as sender address, IP addresses, size, message ID, etc. as CGI parameters. It allows the domain owner to completely hide his network structure and the authorization method, and to implement any arbitrary mechanism. Just as an extreme example to point out the capability, the domain owner's server could calculate a

horoscope on request and decide whether to permit or not based on whether the planet constellation promises the e-mail to be lucky. Obviously, in reality the domain owner would use some kind of database to verify the sender. The domain owner is free to implement anything.

- If the domain owner chooses to use a URL with CGI parameters, to use HTTPS, or to instruct caches to not cache, the server

will be queried for every single e-mail. This allows the server to limit the number of e-mails sent and to detect anomalies, e.g. that a machine has been infected by some worm or virus.

11. Security Considerations

11.1. Draft specific considerations

11.1.1. Authentication strength

It is important to stress, that the suggested method does not provide high level security and does not completely prevent forged e-mails or spam under any circumstances. It is a robust, but not highly reliable and completely secure security mechanism. Keep in mind that it is based on DNS, and DNS is not secure today. Authorization is based on the IP address. The very same machine with the very same IP address could be authorized to send e-mail with a given sender address and sending spam at the same time. Maybe because several users are logged in. Or because several customers use the same relay of the same ISP, where one customer could use the sender address of a different customer. It is up to the ISP to prevent this or not. Machines can still be hijacked. Spammers are also domain owners. They can simply use their own domain and authorize themselves. You will always find people on the world who do not care about security and open their relays and RMX records for others to abuse them. RMX is to be considered as a very cheap and simple light weight mechanism, which can nevertheless provide a significant improvement in mail security against a certain class of attacks, until a successor of SMTP has been defined and commonly accepted.

11.1.2. Where Authentication and Authorization end

Early versions of RMX drafts did not cover the local part of the e-mail address, i.e. what's on the left side of the @ sign. This is still to be discussed. Authentication and authorization are limited to the sending MTA's IP address. The authentication is limited to the TCP functionality, which is sufficient for light weight authentication. The RMX records authorize the IP address of the sending host only, not the particular sender of the message. So if a machine is authorized to use sender addresses of more than a single domain, the authentication scheme does not prevent that any user on this machine can send with any of these domains. RMX is not a substitute for the host security of the involved machines.

The proposed authentication scheme can be seen as a "half way

authentication": It does not track back an e-mail to the effective sender. It tracks only half of the way, i. e. it tracks back to the domain and it's DNS administrators who authorized that particular sender IP address to use it for sending e-mail. How the party responsible for that domain performs user authentication, whom it grants access to, how it helds people responsible for

abuse, is completely left as the private business of those who are in charge of that domain. So this draft does not interfere with the domain's individual security policy or any legislation about such policies. On the other hand, the proposed authentication scheme does not give any statement about the nature and quality of the domain's security policy. This is an essential feature of the proposal: E-mail authentication must be deployed world wide, otherwise it won't do the job. Any security scheme interfering with the local legislations or the domain's security policy will not be accepted and can't effectively deployed. Therefore, the security policy must remain the domain's private business, no matter how lousy the policy might be.

In order to achieve this and to make use of the only existing world wide Internet directory scheme (DNS), the approach of this proposal is to just ignore the local part of the sender address (i.e. what's left of the @ part) and limit view to the domain part. After all, that's what we do anyway when delivering to a given address with SMTP.

11.1.3. Vulnerability of DNS

DNS is an essential part of the proposed authentication scheme, since it requires any directory service, and DNS is currently the only one available. Unfortunately, DNS is vulnerable and can be spoofed and poisoned. This flaw is commonly known and weakens many network services, but for reasons beyond that draft DNS has not been significantly improved yet. Several commentors to previous drafts asked to not use DNS because of its lack of security. This is unfeasible: Any authentication/authorization system linked to some kind of symbolic identity (in this case the domain name) needs some kind of infrastructure and trusted assignment. There are basically two ways to do it: Do it yourself and trust nobody else, or let someone else do it. There are methods to do it the former way, e.g. to give someone some kind of authentication/authorization information after a first successful e-mail exchange, e.g. some kind of cookie or special e-mail address. This is certainly interesting and powerful, but it does not solve the problem on a world wide scale and is far to complicated and error prone for the average user, i. e. 99% of the users.

The latter method to let someone else do the symbolic name assignment and create the authentication framework is well known. It context of public key cryptography, this is called a Public Key

Infrastructure (PKI). One of the best known facts about PKIs is that, until now, we don't have any covering a significant part of the Internet. And we won't have any in near future. The complexity is far too high, it is too expensive, and it involves

cooperation of every single user, which is simply unrealistic and extremely error prone. So what do we have we can use? All we have is the DNS and the Whois database. And we have countries who don't allow cryptography. So the proposal was designed to use DNS without cryptography. It does not avoid DNS because of its vulnerability, it asks for a better DNS, but accepts the DNS as it is for the moment. Currently there are two main threats caused by the DNS weakness:

- A spammer/forger could spoof DNS in order to gain false authorization to send fake e-mails.

- An attacker could spoof DNS in order to block delivery from authorized machines, i. e. perform a Denial of Service attack.

The first one is rather unrealistic, because it would require an average spammer to poison a significant part of the DNS servers of its victims. A spammer sending messages to one million receipients would need to poison at least 1-10% which is 10,000 to 100,000 receipient's DNS servers. This should be unfeasible in most cases.

In contrast, the second threat is a severe one. If an attacker wanted to block messages from one company to another, he just needs to poison the recipients DNS server with a wrong RMX record in order to make the recipient's SMTP machine reject all messages. And this is feasible since the attacker needs to poison only a single DNS server. But does this make SMTP more vulnerable? No. Because the attacker can already do even more without RMX. By poisoning the sender's DNS server with wrong MX records, the attacker can also block message delivery or even redirect the messages to the attacker's machine, thus preventing any delivery error messages and furthermore getting access to the messages.

As a consequence, e-mail delivery by SMTP requires a better DNS anyway. The requirements are not significantly expanded by RMX.

11.1.4. Sneaking RMX attack?

A certain kind of sneaking DNS attack could be possible. DNS and RMX implementors should take care to void it.

Imagine an unauthorized sender is sending a forged mail (e.g. spam). At connection time, before querying the RMX record, the receiving MTA usually performs a PTR query for the IP address of the sending MTA. If the sender has control over the authoritative name server for that particular IP address, the sender could give a normal PTR answer, but could append a wrong RMX, APL, or A record in the additional section of the query. A subsequent RMX query

could receive wrong DNS data if the DNS server used by the receiving MTA accepted those forged records.

11.1.5. Open SMTP relays

Open SMTP relays (i.e. machines which accept any e-mail message from anyone and deliver to the world) abused by spammers are a one of the main problems of spam defense and sender backtracking. In most cases this problem just vanishes because foreign open relay machines will not be covered by the RMX records of the forged sender address. But there are two special cases:

If the spammer knows about a domain which authorizes this particular machine, that domain can be abused for forgery. But in this case, the IP address of the relay machine and the RMX records of the domain track back to the persons responsible. Both can be demanded to fix the relay or remove the RMX record for this machine. An open relay is a security flaw like leaving the machine open for everybody to login and send random mails from inside. Once the administrative persons refuse to solve the problem, they can be identified as spammers and held responsible.

The second special case is when a domain authorizes all IP addresses by having the network 0.0.0.0/0 in the RMX/APL record. In this case, open relays don't make things worse. It's up to the recipient's MTA to reject mails from domains with loose security policies.

11.1.6. Unforged Spam

RMX does not prevent spam (which is, by the way, not yet exactly defined), it prevents forgery. Since spam is against law and violates the recipients rights, spam depends on untracability of the sender. In practice the sender forges the sender address (other cases see below). RMX is designed to detect such forgeries.

However, the RMX approach is rendered ineffective, if the sender does not forge. If the sender uses just a normal address of his own domain, this is just a plain, normal e-mail, which needs to be let through. Since it is up to the human's taste whether this is spam or not, there's no technical way to reliably identify this as

spam. But since the sender domain is known, this domain can be blacklisted or legal steps can be gone into.

11.1.7. Reliability of Whois Entries

Once the RMX infrastructure gets deployed, what's the security gain? It allows to determine the domain which's DNS zone

authorized the sending machine. What's that good for? There are some immediate uses of the domain name, e.g. in black- and whitelisting. But in most cases this is just the starting point of further investigations, either performed automatically before message acceptance, or manually after spam has been received and complainted about.

The next step after determining the domain is determining the people responsible for this domain. This can sometimes be achieved by querying the Whois databases. Unfortunately, many whois entries are useless because they are incomplete, wrong, obsolete, or in uncommon languages. Furthermore, there are several formats of address informations which make it difficult to automatically extract the address. Sometimes the whois entry identifies the provider and not the owner of the domain. Whois servers are not built for high availability and sometimes unreachable.

Therefore, a mandatory standard is required about the contents and the format of whois entries, and the availability of the servers. After receiving the MAIL FROM SMTP command with the sender envelope address, the receiving MTA could check the RMX record and Whois entry. If it doesn't point to a real human, the message could be rejected and an error message like "Ask your provider to fix your Whois entry" could be issued. Obviously, domain providers must be held responsible for wrong entries. It might still be acceptable to allow anonymous domains, i. e. domains which don't point to a responsible human. But it is the receivers choice to accept e-mails from such domains or not.

11.1.8. Hazards for Freedom of Speech

Currently, some governments try to enforce limitations of internet traffic in order to cut unwanted content providers from the network. Some of these governments try to hide a whole country behind firewalls, others try to force Internet providers to poison DNS servers with wrong A records for web servers, e.g. one county administration in Germany tries to do so. If message reception depends on DNS entries, the same governments will try to block not only HTTP, but SMTP from these domains also.

However, since most MTAs already reject messages from unresolvable domain names this is not a new threat.

11.2. General Considerations about spam defense

After discussing security requirements of the proposal, now the security advantages of the RMX approach over content based filters will be explained. Basically, there are three kinds of content

filters:

- Those which upload the message or some digest to an external third party and ask "Is this spam"?

- Those which download a set of patterns and rules from a third party and apply this set to incoming messages in order to determine whether it is spam.

- Those which are independent and don't contact any third party, but try to learn themselves what is spam and what isn't.

The message filters provided by some e-mail service providers are usually not a kind of their own, but a combination of the first two kinds.

11.2.1. Action vs. reaction

Content filters suffer from a fundamental design problem: They are late. They need to see some content of the same kind before in order to learn and to block further distribution.

This works for viruses and worms, which redistribute. This doesn't work for spam, since spam is usually not redistributed after the first delivery. When the filters have learned or downloaded new pattern sets, it's too late.

RMX does not have this problem.

11.2.2. Content based Denial of Service attacks

All three kinds of content filters, but especially the second and the third kind are vulnerable to content based Denial of Service attacks.

If some kind of third party (e.g. non-democratic government, intellectual property warriors, religious groups, military, secret

services, patriots, public relation agents, etc.) wants certain contents not to be distributed, they could either poison the pattern/rule databases or feed wrong sets to particular receivers.

Such pattern/rule sets are the perfect tool for censoring e-mail traffic and denial of service attacks by governments and other parties, and a similar threat are virus filters. E. g. the content industry could demand to teach all virus and spam filters to delete all e-mails containing the URL of an MP3 web server outside the legislation. Software manufacturers could try to block all e-mails

containing software license keys, thus trying to make unallowed distribution more difficult. Governments could try to block distribution of unwanted information and politically incorrect speech.

RMX does not have this problem.

12. Privacy Considerations

(It was proposed on the 56th IETF meeting to have a privacy section in drafts and RFCs.)

12.1. Draft specific considerations

12.1.1. No content leaking

Since the RMX approach doesn't touch the contents of a message in any way, there is obviously no way of leaking out any information about the content of the message. RMX is based solely on the envelope recipient address. However, methods to fix problems not covered by RMX might allow content leaking, e.g. if the acceptance of a message with an empty sender address requires the reference to the message id of an e-mail recently sent, this allows an attacker to verify whether a certain message was delivered from there.

12.1.2. Message reception and sender domain

Message delivery triggers RMX and APL requests by the recipient. Thus, the admin of the DNS server or an eavesdropper could learn that the given machine has just received a message with a sender from this address, even if the SMTP traffic itself had been encrypted.

However, most of today's MTAs do query the MX and A records of the domain after the MAIL FROM command, so this is not a real new threat.

12.1.3. Network structure

Since RMX and its associated APL records provide a complete list of all IP addresses of hosts authorized to send messages from this address, they do reveal informations about the network structure and maybe the lifestyle of the domain owner, since a growing number of domains are owned by single persons or families. E.g. the RMX records could reveal where someone has his job or spends his time at weekends.

If such informations are to be kept secret, it is the user's job to not sent e-mails from there and to relay them from non-compromising IP addresses.

12.1.4. Owner information distribution

As described above, RMX depends partly on the reliability of the whois database entries. It does not make anonymous domains

impossible, but it requires to keep the database entries "true", i. e. if a whois entry does not contain informations about the responsible person, this must be unambigously labeled as anonymous. It must not contain fake names and addresses to pretend a non-existing person. However, since most Internet users on the world feel extremely annoyed by spam, they will urge their MTA admin to reject messages from anonymous domains. The domain owner will have the choice to either remain anonymous but be not able to send e-mail to everyone in the world, or to be able but to reveal his identity to everyone on the world.

It would be possible to provide whois-like services only to recipients of recent messages, but this would make things too complicated to be commonly adopted.

12.2. General Considerations about spam defense

12.2.1. Content leaking of content filters

As described above in the Security chapter, there are spam filters which inherently allow leakage of the message body. Those filters upload either the message body, or in most cases just some kind of checksum to a third party, which replies whether this is to be seen as spam or not. The idea is to keep a databases of all digests of all messages. If a message is sent more often than some threshold, it is to be considered as a mass mail and therefore tagged as spam.

While the digest itself does not reveal the content of the message, it perfectly reveals where a particular message has been delivered to. If a government finds just a single unwanted message, if a software manufacturer finds a single message with a stolen product license key, if someone finds a message with unpatriotic content, it takes just a single database lookup to get a list of all people who received this particular message. Content filters with digest upload are "Big Brother's" favourite toy.

12.2.2. Black- and Whitelists

Some proposals against spam are based on a central database of white- or blacklisted IP addresses, Sender names, Message IDs or whatever. Again, there is a central database which learns who has

received which e-mail or from which sender with every query. This allows tracking relations between persons, which is also a breach of privacy.

13. Deployment Considerations

13.1. Compatibility

13.1.1. Compatibility with old mail receivers

Since the suggested extension doesn't change the SMTP protocol at all, it is fully compatible with old mail receivers. They simply don't ask for the RMX records and don't perform the check.

13.1.2. Compatibility with old mail senders

Since the SMTP protocol is unchanged and the SMTP sender is not involved in the check, the method is fully compatible with old mail senders.

13.1.3. Compatibility with old DNS clients

Since the RMX is a new RR, the existing DNS protocol and zone informations remain completely untouched.

If RMX is provided as a TXT record instead, it must be ensured that no other software is misinterpreting this entry.

13.1.4. Compatibility with old DNS servers

Full compatibility: If the server does not support RMX records, RMX in TXT records can be used.

13.2. Enforcement policy

Obviously, for reasons of backward compatibility and smooth introduction of this scheme, RMX records can't be required immediately. Domains without RMX records must temporarily be treated the same way as they are treated right now, i.e. e-mail must be accepted from anywhere. But once the scheme becomes sufficiently widespread, mail relays can start to refuse e-mails

with sender addresses from domains without RMX records, thus forcing the owner of the domain to include a statement of authorization into the domain's zone table. Domain owners will still be free to have an RMX record with a network and mask 0.0.0.0/0, i.e. to allow e-mails with that domain from everywhere. On the other hand, mail receivers will be free to refuse mails from domains without RMX records or RMX records which are too loose. Advanced MTAs might have a configuration option to set the maximum number of IP addresses authorized to use a domain. E-mails from a domain, which's RMX records exceed this limit, would be rejected. For example, a relay could reject e-mails from domains which

authorize more than 8 IP addresses. That allows to accept e-mails only from domains with a reasonable security policy.

14. General considerations about fighting spam

Is there a concise technical solution against spam? Yes.

Will it be deployed? Probably not.

Why not? Because of the strong non-technical interests of several parties against a solution to the problem, as described below. Since these are non-technical reasons, they might be beyond the scope of such a draft. But since they are the main problems that prevent fighting spam, it is unavoidable to address them.

14.1. The economical problem

As has been recently illustrated in the initial session of the IRTF's Anti Spam Research Group (ASRG) on the 56th IETF meeting, sending spam is a business with significant revenues.

But a much bigger business is selling anti-spam software. This is a billion dollar market, and it is rapidly growing. Any simple and effective solution against spam would defeat revenues and drive several companies into bankrupt, would make consultants jobless.

Therefore, spam is essential for the anti-spam business. If there is no spam, then no Anti-Spam software can be sold, similar to the anti-virus business. There are extremely strong efforts to keep this market growing. Viruses, Worms, and now spam are just perfect to keep this market alive: It is not sufficient to just buy a software. Databases need to be updated continuously, thus making the cash flow continuously. Have a single, simple, and permanent solution to the problem and - boom - this billion dollar market is dead. That's one of the reasons why people are expected to live with spam. They have to live with it to make them buy anti-spam software. Content filters are perfect products to keep this market alive.

14.2. The POP problem

Another problem is the history of mail delivery. Once upon a time,

there used to be very few SMTP relays which handled the e-mail traffic of all the world, and everybody was happy with that. Then odd things like Personal Computers, which are sometimes switched off, portable computers, dynamicaly assigned IP addresses, IP access from hotel rooms, etc. was invented, and people became unhappy, because SMTP does not support delivery to such machines. To make them happy again, the Post Office Protocol[9] was invented, which turned the last part of message delivery from SMTP's push

style into a pull style, thus making virtually every computer on the world with any random IP address a potential receiver of mails for random domains. Unfortunately, only receiving e-mail was covered, but sending e-mail was left to SMTP.

The result is that today we have only very few SMTP relays pointed to by MX records, but an extreme number of hosts sending e-mail with SMTP from any IP address with sender addresses from any domain. Mail delivery has become very asymmetric. Insecurity, especially forgeability, has become an essential part of mail transport.

That problem could easily be fixed: Use protocols which allow uploading of messages to be delivered. If a host doesn't receive messages by SMTP, it shouldn't deliver by SMTP. Mail delivery should go the same way back that incoming mail went in. This is not a limitation to those people on the road who plug their portable computer in any hotel room's phone plug and use any provider. If there is a POP server granting download access from anywhere, then the same server should be ready to accept uploading of outgoing messages.

But as comments on the first draft version of this RFC showed, people religiously insist on sending e-mail with their domain from any computer with any IP address in the world, e.g. when visiting a friend using her computer. It appears to be impossible to convince people that stopping mail forgery requires every one of them to give up forging.

14.3. The network structure problem

A subsequent problem is that many organisations failed to implement a proper mail delivery structure and heavily based their network on this asymmetry. The author received harsh comments from Universities who were unable to give their network a reasonable structure. While they do have a central mail relay for incoming mail to the universities domain, they developed a structure where every member of the University randomly sends e-mails with that University's domain as a sender address from home or everywhere in the world with any dynamically assigned IP address from any provider. So this domain is to be used from every possible IP address on earth, and they are unable to operate any authentication scheme. Furthermore, they were unable to understand that such a

policy heavily supports spam and that they have to expect that people don't accept such e-mails anymore once they become blacklisted.

As long as organisations insist on having such policies, spammers

will have a perfect playground.

14.4. The mentality problem

Another problem is the mentality of many internet users of certain countries. The author received harsh comments from people who strongly insisted on the freedom to send any e-mail with any sender address from anywhere, and who heavily refused any kind of authentication step or any limitation, because they claimed that this would infringe their constitutional "Freedom of Speech". They are undeviatingly convinced that "Freedom of Speech" guarantees their right to talk to everybody with any sender address, and that is has to be kept the recipient's own problem to sort out what he doesn't want to read - on the recipient's expense. The author learned that it is extremely difficult to convince some people to give up random e-mail sending. However, a security mechanism for the world wide mail system can never meet the taste and the requirements of every single one of all those hundreds of millions of users.

It requires a clear statement that the constitutional "Freedom of Speech" does not cover molesting people with unsolicited e-mail with forged sender address.

14.5. The identity problem

How does one fight against mail forgery? With authentication. What is authentication? In simple words: Making sure that the sender's real identity meets the recipients idea of who is the sender, based on the sender address which came with the message.

What is identity? It is the main problem. Several countries have different ideas of "identity", which turn out to be somehow incompatible. In some countries people have identity cards and never change their name and birthday. Identities are created by human birth, not by identity changes. Other countries do not have such a tight idea about identity. People's temporary identity is based on nothing more than a driving license and a social security number. With this background, it is virtually impossible to create a trustworthy PKI covering all Internet users.

14.6. The multi-legislation problem

Many proposals about fighting spam are feasible under certain legislations only, and are inacceptable under some of the legislations. But a world wide applicable method is required. That's why the approach to ask everone on the world to sign messages with cryptographic keys is not feasible.

References

1. **J. Klensin, "Simple Mail Transfer Protocol," RFC 2821 (April 2001).**

2. **P. Resnick, "Internet Message Format," RFC 2822 (April 2001).**

3. **P. Koch, "A DNS RR Type for Lists of Address Prefixes (APL RR),"** RFC 3123 (June 2001).

4. **T. Dierks, C. Allen, "The TLS Protocol," RFC 2246 (January 1999).**

5. **J. Myers, "Simple Authentication and Security Layer (SASL),"** RFC 2222 (October 1997).

6. **P. Mockapetris, "DOMAIN NAMES - IMPLEMENTATION AND SPECIFICATION,"** RFC 1035 (November 1987).

7. **T. Berners-Lee and others, "Hypertext Transfer Protocol HTTP/1.1,"** RFC 2616 (June 1999).

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

9. **J. Myers, M. Rose, "Post Office Protocol - Version 3," RFC 1939** (May 1996).

Draft History

00 Dec 2002
01 Apr 2003
02 Jun 2003
03 Oct 2003

Author's Address

Hadmut Danisch

Tennesseeallee 58
76149 Karlsruhe
Germany

Phone: ++49-721-843004 or ++49-351-4850477
E-Mail: rfc@danisch.de

Comments

Please send comments to rfc@danisch.de.

Expiry

This drafts expires on Nov 1, 2004.

Network Working Group
Request for Comments: nnnn
Category: Experimental

H. Danisch

May 2004

The RMX DNS RR and method for lightweight SMTP sender authorization

Status of this Memo

This memo provides information for the Internet community. It does not specify an Internet standard of any kind. Distribution of this memo is unlimited.

Copyright Notice

Abstract

This memo introduces a new authorization scheme for SMTP e-mail transport. It is designed to be a simple and robust protection against e-mail fraud, spam, and worms. It is based solely on organisational security mechanisms and does not require but still allow use of cryptography. This memo also focuses on security and privacy problems and requirements in context of spam defense.

This document is part of the LMAP work of the Anti-Spam Research Group (ASRG) of the IRTF.

Table of Contents

1. Problem and threat description

1.1. Mail sender forgery

The amount of e-mails with forged sender addresses has dramatically increased. As a consequence, damages and annoyances caused by such e-mails increased as well. In the majority of examined e-mails the domain name of the envelope sender address was forged, and the e-mail was sent from an IP address which does not belong to a network used by the actual owner of the domain.

1.1.1. Definition of sender forgery

As discussions, comments to prior drafts of this RFC, and different approaches to stop forgery showed, different perceptions of "mail forgery" exist. For example, there are mechanisms to verify e-mail addresses for mailing lists, web servers, or to stop spam, which do send a message with a random number to the given address and expect the user to send a reply. Here, someone is considered to be allowed to use a particular e-mail address, if and only if he is able to receive messages sent to this address, and is able to reply to such a message. While this definition appears to be quite plausible and natural, it can't be used for a simple technical solution. Sending back a challenge and expecting a reply is simply too much overhead and time delay, and not every authorized sender is able and willing to reply (e.g. because he went offline or is not a human).

Within the scope of this memo, sender forgery means that the initiator of an e-mail transfer (which is the original sender in contrast to relays) uses a sender address which he was not authorized to use. Being authorized to use an address means that the owner (administrator) of the internet domain has given permission, i.e. agrees with the use of the address by that particular sender. This memo will cover both the permission of the full e-mail address and the domain part only for simplicity.

Within context of Internet and SMTP, the sender address usually occurs twice, once as the envelope sender address in SMTP [1], and once as the address given in the mail header [2]. While the following considerations apply to both addresses in principle, it is important to stress that both addresses have distinct semantics

and are not necessarily the same. The envelope address identifies the initiator of the transport, while the header identifies the author of the message content. Since this memo deals with the message transport only and completely ignores the message content, the method should naturally be applied to the envelope sender address. However, this is currently under discussion in the ASRG

and the IETF working groups.

1.1.2. Spam

A common and well known problem is the dramatic increase of unsolicited e-mail, commonly called "spam". Again, the majority of examined e-mails had forged sender addresses. The abused domains were mainly those of common webmailers as hotmail or yahoo, or well-known companies.

Unfortunately, there is no accurate definition of spam available yet, and neither are there concise technical criterions to filter or block spam with technical mechanisms. There are efforts to design content based filters, but these filters are expensive in calculation time (and sometimes money), and they do not reliably produce predictable results. They usually give false positives and/or require user interaction. Content filters in general suffer from a design problem described later in this memo. Therefore, this proposal does not use the content based approach to block spam.

As analysis of spam messages showed, most of spam messages were sent with forged envelope sender addresses. This has mainly three reasons. The first reason is, that spam senders usually do not want to be contacted by e-mail. The second reason is, that they do not want to be blacklisted easily. The third reason is, that spam is or is going to be unlawful in many countries, and the sender does not want to reveal his identity. Therefore, spam is considered to be a special case of sender forgery throughout this memo.

1.1.3. E-Mail Worms

Another example of sender forgery is the reproduction of e-mail worms. Most worms use random sender addresses, e.g. the addresses found in mailboxes on the infected system. In most cases analyzed by the author, the e-mails sent by the reproduction process can also be categorized as forged, since the infected system would under normal circumstances not be authorized to send e-mails with such e-mail addresses. So forgery does not require a malicious human to be directly involved. This memo covers any kind of e-mail sender address forgery, included those generated by malicious

software.

1.1.4. E-Mail spoofing and fraud

Forging e-mail sender addresses for fraud or other kinds of deception ("human engineering") has also dramatically increased.

There are many known cases where single or mass e-mails were sent with false sender addresses, pretending to come from service providers, software manufacturers etc., and asking the receiver to install any software or patches, or to reply with any confidential information. The Internet is increasingly becoming a scene of crime, and so are it's services, including e-mail. It is obvious that crime based on e-mail is eased by the fact that SMTP allows arbitrary sender address spoofing.

1.2. Indirect damage caused by forgery

As observed by the author, mass mails and worms with forged sender addresses can cause a severe damage for the real owner of the abused sender addresses. If a sender A is sending an e-mail to the receiver B, pretending to be C by using a sender address of C's domain, then C has currently no chance to prevent this, since C's machines and software are not involved in any way in the delivery process between A and B. B will nevertheless send any error messages (virus/spam alert, "no such user", etc.) to C, erroneously assuming that the message was sent by C. The author found several cases where this flood of error messages caused a severe denial of service or a dramatic increase of costs, e.g. when C was downloading the e-mail through expensive or low bandwidth connections (e.g. modem or mobile phones), or where disk space was limited. The author examined mass mailings, where several tens or hundreds of thousands of messages were sent to recipients around the world, where these messages caused only annoyance. But since several thousands of these recipient addresses were invalid or didn't accept the message, the owner of the DNS domain which was abused by the spammer to forge sender addresses was flooded for several months with thousands of error messages, jamming the e-mail system and causing severe costs and damages.

As a consequence, when A sends a message to B, pretending to be C, there must be any mechanism to allow C to inform B about the fact, that A is not authorized to use C as a sender address. This is what this memo is about.

1.3. Technical problem analysis

Why does e-mail forgery actually exist? Because of the lack of the Simple Mail Transfer Protocol SMTP[1] to provide any kind of sender authentication, authorization, or verification. SMTP was designed

at a time where security was not an issue. Efforts have been made to block forged e-mails by requiring the domain part of the sender address to be resolvable. This method provides protection from e-mails with non-existing sender domains, and indeed, for some time it blocked most spam e-mails. However, since attackers and spam

senders began to abuse existing domain names, this method was rendered ineffective.

1.4. Shortcomings of cryptographical approaches

At a first glance, the problem of sender address forgery might appear to be solvable with cryptographical methods such as challenge response authentications or digital signatures. A deeper analysis shows that only a small, closed user group could be covered with cryptographical methods. Any method used to stop spam forgery must be suitable to detect forgery not only for a small number of particular addresses, but for all addresses on the world. An attacker does not need to know the secrets belonging to a particular address. For him it is sufficient to be able to forge any address and thus to know any secret key. Since there are several hundreds of millions of users, there will always be a large amount of compromised keys, thus spoiling any common cryptographic method. Furthermore, cryptography has proven to be far too complicated and error prone to be commonly administered and reliably implemented. Many e-mail and DNS administrators do not have the knowledge required to deal with cryptographic mechanisms. The most important requirement for a world wide applicable spam protection is simplicity. Many legislations do not allow the general deployment of cryptography and a directory service with public keys. For these reasons, cryptography is applicable only to a small and closed group of users, but not to all participants of the e-mail service.

After all, after more than 20 years of Public Key Cryptography, there is still no common Public Key Infrastructure, there is still not enough adequate crypto software available, neither hardware devices, and existing crypto software is far from being robust and free of severe bugs. Cryptography cannot be expected to solve the spam problem in the foreseeable future.

2. A DNS based sender address verification

2.1. Overview

To gain an improvement in e-mail authenticity while keeping as much SMTP compatibility as possible, a method is suggested which doesn't change SMTP at all.

The idea is to store the information about how to verify who is authorized to transmit e-mails through SMTP with a particular sender address (either full address or - for simplicity - only the domain part of the address) in a directory service. The internet's directory service is currently DNS. To be precise, the verification consists of two steps, the classical pair of authentication and authorization:

The first step is the authentication. While several methods are possible to perform authentication (see below), the most important and robust method is the verification of the sender's IP address. This is done implicitely by TCP/IP and the TCP sequence number. The authenticated identity is the IP address. It has to be stressed that this TCP/IP "authentication" is a weak authentication and vulnerable to several attacks. It is nevertheless sufficient for this purpose, especially for blocking spam. It doesn't take any implementation and it doesn't cost: It is already there, it is a functionality of TCP/IP. An incoming SMTP connection based on TCP/IP already carries the sender's IP address without any modification of SMTP. See below (section Entry types) for more details about authentication methods.

The second step is the authorization. It is based on the identity given by the previous authentication step, e.g. the IP address of the originator of the incoming SMTP connection, and on the envelope sender address. The mechanism proposed in this memo answers the question "Is that particular sender (IP address,...) allowed to send with that sender address" by querying and processing authorization records stored in a directory service, which is DNS.

When the sender has issued the "MAIL FROM:" SMTP command, the receiving mail transfer agent (MTA) can - and modern MTAs do - perform some authorization checks, e.g. run a local rulebase or

check whether the sender domain is resolvable.

The suggested method is to let the DNS server for the sender domain provide informations about who - this means for example which IP address - is authorized to use an address or a domain as a part of it. After receiving the "MAIL FROM:" SMTP command, the receiving

MTA can verify, whether e. g. the IP address of the sending MTA is authorized to send mails with this domain name. Therefore, a list of entries with authorized IP addresses or other descriptions is provided by the authoritative DNS server of that domain. The entry types are described in the subsequent chapters. Some of these entry types are

- An IPv4 or IPv6 network address and mask
- A fully qualified domain name referring to an A record
- A fully qualified domain name referring to an APL record

RMX records of these types would look like this:

```
somedomain.de.      IN RMX ipv4:10.0.0.0/8
rmxtest.de.         IN RMX host:relay.anyprovider.com
danisch.de.         IN RMX apl:relays.rackland.de
relays.rackland.de. IN APL 1:213.133.101.23/32 1:1.2.3.0/24
```

where the machine with the example address 213.133.101.23 and the machines in the example subnet 1.2.3.0/24 are the only machines allowed to send e-mails with an envelope sender address of domain danisch.de. Since the APL records do not necessarily belong to the same domain or zone table as the RMX records, this easily allows to refer to APL records defined by someone else, e.g. the internet access or server hosting provider, thus reducing administrative overhead to a minimum. In the example given above, the domain danisch.de and several other domains are hosted by the service provider Rackland. So if the relay structure of Rackland is modified, only the zone of rackland.de needs to be updated. The domain owners don't need to care about such details.

2.2. Envelope vs. header sender address

Questions were raised why the proposed mechanism is based on the envelope sender address, and not on the sender address given in the message header. Technically, both can be used. Actually, it makes sense to use the envelope address.

In common, the header sender address identifies the author of the content, while the envelope sender tells who caused the transmission. The approach proposed in this memo is transmission

based, not content based. We can not authorize the author of a message if we don't have contact with him, if the message does not already contain a signature. In contrast, the sending MTA is linked to an IP address which can be used for authentication. This mechanism might not be very strong, but it is available and sufficient to solve today's e-mail security problems.

Some people argued that it is the header address and not the sender address, which is displayed in common mail readers (MUAs), and where the receiver believes the mail to come from. That's true, but it doesn't help. There are many cases where the header sender differs from the envelope sender for good reasons (see below in the consequences chapter for the discussion about relaying). Relaying, mailing lists etc. require to replace the sender address used for RMX. If this were the header address, the message header would have to be modified. This is undesirable.

2.3. Domain part vs. full sender address

Early draft versions of this memo were limited to the domain part of the sender address. The first reason is that it is common and MX-like, to lookup only the domain part of an e-mail address in DNS. The second reason is, that it was left to the private business of the domain administration to handle details of user verification. The idea was that the domain administration takes care to verify the left part of an e-mail address with an arbitrary method of their individual taste. RMX was originally designed to ignore the left part of the address and to expect the domain administration to take over responsibility for enforcing their policy. If, e.g., a spam message arrived and passed the RMX mechanism, it is known to be authorized by the domain administration and they can be blamed, no matter what is on the left side of the sender address - it's their private problem what happens on the left side of the @. By far the most of the comments to prior draft versions of this memo agreed with that. A few comments asked for a finer granularity.

And indeed, there is no technical reason against a finer granularity. All it takes is a mapping from a given envelope sender address to a DNS name, and the RMX lookup for that particular e-mail address could be done instead of a lookup for the domain part only. However, to my knowledge, most domain administrators would not like to provide an RMX entry for every single e-mail address. In many cases, this would also overload DNS servers.

It is to be discussed how to cover both views. One method could be to query the full address, and if no RMX records were found to query the domain part only. A different approach would be to query the domain part only, and if it's RMX record contains a special entry, then a new query for the full address is triggered. A third

way would be to always query the full address and to leave the problem to the wildcard mechanism of DNS.

A completely different approach to allow authorization with full

address and even much finer granularity is the RMX++ proposal mentioned in the future development section below.

3. Mapping of E-Mail addresses to DNS names

To perform the RMX query, a mapping is needed from E-Mail addresses to DNS fully qualified domain names. In other words: A function is needed which tells for every incoming e-mail where in DNS to look for authorization records.

This chapter is only a rough outline. Details are currently under discussion in the ASRG and IETF working groups.

3.1. Domain part only

Mapping of the domain part is trivial, since the domain part of an e-mail address itself is a valid DNS name and does not need translation. It might be nevertheless desirable to distinguish the RMX entries from other entries, depending of the encoding of the records. If the RMX entries are encoded in TXT record types, they might collide with other uses of TXT records. It might be necessary to prepend the domain part with a special prefix, e.g. _rmx. So the e-mail address some.user@example.com could be mapped to example.com or _rmx.example.com.

3.2. Full address

Mapping a full address is slightly more difficult. The @ symbol must be unambiguously translated, and therefore can not be simply translated into a dot. The e-mail addresses some.user@example.com and some@user.example.com must have different mappings. Therefore, the @ symbol could be translated into _rmx, implicitely assuming that this is not an allowed domain name component of normal domain names. Then the rightmost _rmx in the mapped DNS name always corresponds to the @ symbol. some.user@example.com would be translated into some.user._rmx.example.com and can be covered by a wildcard entry like *._rmx.example.com.

Character encoding and character sets are still to be discussed.

3.3. Empty address

Unfortunately, SMTP allows empty envelope sender addresses to be

used for error messages. Empty sender addresses can therefore not be prohibited. As observed, a significant amount of spam was sent with such an empty sender address. To solve this problem, the host name given in the HELO or EHLO command could be used instead to lookup the RMX records. This makes sense, since such messages were generated by the machine, not a human.

4. Mandatory entry types and their syntax

The entry types described in this section MUST be supported by all implementations of this memo.

4.1. Overall structure

Similar to APL, an RMX record is just a concatenation of zero or more RMX entries. The entries within one record form an ordered rule base as commonly usual in packet filtes and firewall rulesets, i. e. they are processed one ofter another until the first entry matches. This entry determines the result of the query. Once a matching entry is found, the RMX processing is finished.

For any domain name there should not exist more than a single RMX record. Due to the structure of DNS, it is nevertheless possible to have more than a single RMX record. Multiple RMX records are treated as a single record consisting of the concatenation of all records. While the entries in a record are ordered, the records are not ordered and may be processed in arbitrary order. If the order of the entries matters, it is the zone maintainer's responsibility to keep those entries in a single record. For example, there are negative entries, which exclude IP addresses from authorization. It is important that these entries are processed before positive entries giving permission to a wider address range. Since order is guaranteed only within a record, corresponding negative and positive entries must be put in the same record.

An RMX record may consist of one or more entries, where the entries are separated by whitespace. An entry must not contain white space. Each entry consists of an optional exclamation sign, a tag, a colon, and the entry data:

```
[!] TAG : ENTRY-SPECIFIC-DATA
```

If the entry starts with an exclamation sign, the entry is negated. See the entry type description below for details.

The TAG is the mnemonic type identifier or the decimal number of

the entry. The TAG is case-insensitive. It is immediately followed by a colon.

The syntax and semantics of ENTRY-SPECIFIC-DATA depends of the the entry type. See description below.

Example:

```
danisch.de.  IN RMX apl:relays.rackland.de !ipv4:1.2.3.5
                    ipv4:1.2.3.0/24
```

4.2. Unused

This is a primitive entry which just says that this sender address will never be used as a sender address under any circumstances. Example:

```
testdomain.danisch.de    IN RMX unused:
```

4.3. IPv4 and IPv6 address ranges

These entry types contain a bit sequence representing a CIDR address part. If that bit sequence matches the given IP address, authorization is granted or denied, depending on the negation flag.

The entry is prepended with the tag "IPv4" or "IPv6". The colon is followed with an IPv4 or IPv6 address in standard notation, optionally followed by a slash and a mask length. If the negation flag is set, then the given address range is excluded. Examples:

```
danisch.de   IN RMX ipv4:213.133.101.23 ipv6:fe00::0
             IN RMX ipv4:10.0.0.0/8     ipv6:fec0::0/16
             IN RMX !ipv4:1.2.3.4
```

(Please note that it does not make much sense to use RFC1918-Addresses in RMX records, this is just to give a syntax example.)

4.4. DNS Hostnames and Dynamic IP addresses

This entry type simply contains a regular DNS name, which is to be resolved as a host name (fetch the A record or IPv6 equivalent). If the given IP address matches the result, authorization is granted or denied, depending on the negation flag.

The entry is prepended with the tag "host", followed by a colon and

the hostname. Examples:

```
danisch.de  IN RMX host:relay.provider.de
            IN RMX !host:badmachine.domain.de apl:relays.domain.de
```

Several people argued against RMX that it would break their existing installation which delivers e-mail from dynamically assigned IP addresses, because their IP providers didn't assign a static address, or because they are road warriors, plugging their notebook in any hotel room on the world.

RMX provides a simple solution: If such a machine has a dynamically updated DNS entry (e.g. DynDNS), all it takes is an RMX entry of the hostname type pointing to this dynamic DNS entry.

The cleaner solution would be to deliver mail the same way as it is received: If downloaded by POP from a central relay with a static address, where the MX points to, then it would be a good idea to deliver e-mail the same way in reverse direction. Unfortunately, plain POP does not support uploading yet.

4.5. APL Reference

This entry type simply contains a regular DNS name, which is to be resolved as an APL record [3] index (fetch the APL record). If the given IP address positively matches the APL, authorization is granted. Details of the semantic (espially when the negation bit is set) are still to be defined. It is still to be defined how to treat unresolvable entries.

The entry is prepended with the tag "host", followed by a colon and the hostname. Example:

```
danisch.de     IN RMX apl:relays.rackland.de
```

4.6. Domain Member

In many cases it is desirable to cover all hosts of a given domain with an RMX record without the need to duplicate the list of these hosts. This entry type does it (thanks to Eric A. Hall for pointing out this entry type). It contains a regular DNS name.

If this entry type is given, a reverse DNS query for the IP address of the sending MTA is performed to find its official fully qualified domain name. To prevent spoofing, this domain name is accepted only if a subsequent address query to the given domain name points to exactly the IP address of the sending MTA (the usual procedure to verify PTR records).

The entry matches if the fully qualified domain name of the sending MTA ends in the given domain. The negation flag works as usual.

The tag for this entry type is "domain". After the colon the domain name is given, but might be empty, thus pointing to itself. Example:

```
somedomain.org  IN RMX domain:somedomain.org domain:provider.com
```

would authorize all machines which's hostname can be verified

through an PTR and A query, and which ends in "somedomain.org" or "provider.com".

With such an entry, large companies with different networks can easily be covered with just a single and simple RMX entry. Obviously, it requires proper PTR records.

As a special shortcut, the DNS name may be empty. In this case the domain name of the zone itself is taken. Thus, with a very simple entry of the type

```
somecompany.com  IN RMX domain:
```

a company could authorize all machines which's IP addresses map to DNS names end in somecompany.com, which applies in the majority of companies.

Thus, a simple entry of the form

```
@ IN RMX domain:
```

would be a good starting point for company networks and would in most cases allow easy and simple RMX configuration if the network can't be described with a simple network mask.

4.7. Full Address Query

As described above, RMX records will in most cases apply to the domain part of the sender address. In special cases it might be desirable to query the RMX record for a particular address. An RMX entry of the Full Address Query type may occur in a domain RMX record only. It signals that the RMX record for the full address is to be fetched and processed.

This entry type does not take arguments. The negation flag is not supported. The tag is "full".

If such a full address query is to be performed, the mail address must be mapped to a valid and non-ambiguos DNS name. This mapping is still to be defined. It is not sufficient to simply replace the @ with a dot, because of case sensitivity, character sets, etc. The e-mail addresses

```
john.doe@example.org
John.Doe@example.org
john@doe.example.org
```

must all be mapped to different DNS entries. A better approach is RMX++ (see below).

4.8. MX reference

This entry type has no parameters. It means that all those machines are authorized, which are pointed to by an MX record.

Example:

danisch.de. IN RMX MX:

would simply allow all machines receiving mails for danisch.de (i.e. the MX machines) to deliver as well.

5. Optional and experimental entry types

The following subsections roughly describe further experimental entry types. These methods are just considerations about what to include in RMX and what to not include. The main purpose of this section is to start a discussion about such entry types.

The disadvantage of the following methods is that they violate the basic idea of RMX, i. e. to be simple, robust, easy to implement and easy to administer. The author does not believe that it is a good idea or even feasible to implement cryptography for a world wide e-mail transfer network. Keep in mind that cryptographic keys can be copied. Even if only around 0.01% of the cryptographic keys are stolen, this still compromises and spoils RMX. Cryptography is simply the wrong tool for the problem RMX is intended to solve. It is nevertheless to be discussed.

5.1. TLS fingerprint

The sender is considered to be authorized if the message was transmitted through SMTP and TLS[4], and the sender used a certificate matching the fingerprint given in the RMX record.

5.2. TLS and LDAP

The receiver could perform an LDAP query for the sender address (through the LDAP SRV record or given in the RMX record), fetch the X.509 certificate for the sender. The sender is considered to be authorized when the message was transmitted through SMTP and TLS using this certificate.

5.3. PGP or S/MIME signature

It would be possible to accept a message only if it was signed with PGP or S/MIME with a key which's fingerprint is given in the RMX record or to be fetched from LDAP or any PGP database. This is just for discussion, since it violates the idea of RMX to focus on the transport, not on the content. It would also allow replay attacks and not cover the envelope sender address or message header.

5.4. Transparent Challenge/Response

It would also be possible to implement a challenge-response mechanism without modifying the syntax of SMTP. For example, the receiving MTA could issue a challenge with it's very first greeting message, the sending MTA could include the response in the HELO or EHLO parameter and when the receiving MTA later learns the sender

envelope address, it could verify the response based on entries in the RMX record.

5.5. SASL Challenge/Response

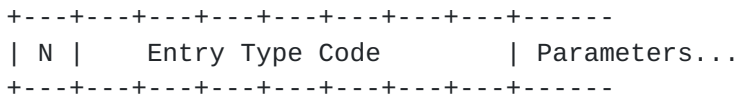
Modern SMTP implementations already include a SASL[5] mechanism, which easily allows to plugin new authentication mechanisms. While common SASL mechanisms require to use a previously shared password, a new mechanism could perform a challenge response authentication as a SASL method.

6. Encoding

6.1. RMX Records

6.1.1. Overall structure

Each entry starts with an octet containing the entry type and the negation flag:



N — If this bit (MSB) is set, an IP address matching this entry is not authorized, but explicitely rejected. See entry type descriptions for details.

Entry Type — A 7bit number simply determining the entry type.

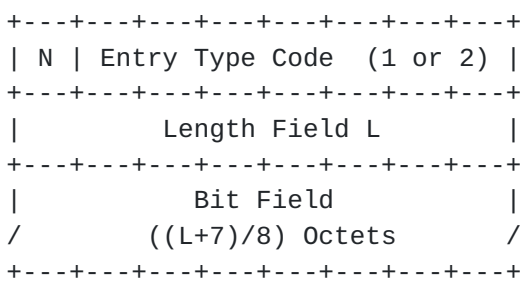
Currently, entries do not have an explicit length field, the entry length is determined implicitely by the entry type. Applications are required to abort if an unknown entry type is found, instead of skipping unknown entries.

6.1.2. Record encoding

A RMX record is simply a concatenation of RMX entries.

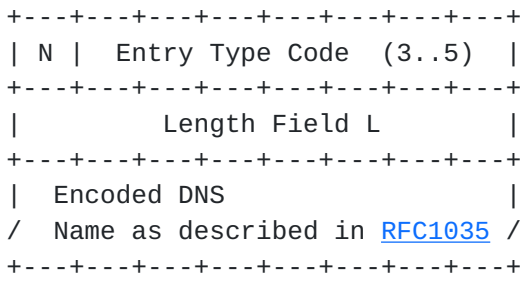
6.1.3. Encoding of IPv4 and IPv6 address ranges

After the entry type tag as described above, one octet follows giving the length L of the bit sequence. Then a sequence of exactly as many octets follows as needed to carry L bits of information (= trunc((L+7)/8)).


| N | Entry Type Code (1 or 2) |
| Length Field L |
| Bit Field |
/ ((L+7)/8) Octets /

6.1.4. Encoding of DNS

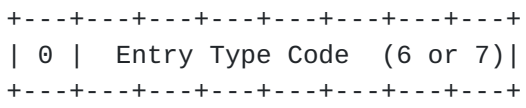
After the entry type tag immediately follows a DNS encoded[6] domain name.



In contrast to earlier draft versions of this memo, the DNS name cannot be compressed, since this would cause decompression errors when a DNS server which does not know this particular RR type is part of the query chain.

6.1.5. Encoding of unused and full address query

These entries do not contain parameters and does not allow the negation flag. So the encoding is quite simple:



6.1.6. Additional Records

In order to avoid the need of a second query to resolve the given host name, a DNS server should enclose the A record for that domain name in the additional section of the additional section of the DNS reply, if the server happens to be authoritative.

In order to avoid the need of a second query to resolve the given host name, a DNS server should enclose the APL record for that domain name in the additional section of the additional section of

the DNS reply, if the server happens to be authoritative.

6.2. Alternative encoding as TXT records

The main objection against the prior versions of this draft was that it requires a new RR entry type and upgrading all DNS servers.

Therefore and alternative encoding is proposed. Instead of using a new RR type, the TXT record type is used to contain the RMX record. The records would simply look as described in the entry type chapters above, e.g.

```
_rmx.danisch.de.   IN TXT "apl:relays.rackland.de"
```

To allow smooth introduction of RMX without the need to immediately upgrade all DNS servers, all clients (which have to be newly installed anyway) MUST support both the TXT and the RMX records. A client has to perform an ANY or a TXT and a RMX query. Servers/zone tables may currently use TXT entries but SHOULD use RMX entries in future.

7. Message Headers

An RMX query must be followed by any kind of action depending on the RMX result. One action might be to reject the message. Another action might be to add a header line to the message body, thus allowing MUAs and delivery programs to filter or sort messages.

In future, the RMX result might be melted into the Received: header line [2].

The details of such entries are to be discussed. As a proposal the following form is suggested:

X-RMX: RESULT addr ADDRESS by HOST on DATE mechanism MECHANISM

where

RESULT is one of "Granted", "Denied", "NotInRMX", "NoRMX", "TempFail", "BadData", "Trusted".

ADDRESS is the IP address of the sending machine

HOST is the name of the machine performing the RMX query.

DATE is the date of the query.

MECHANISM is the RMX method used to authorize the sender.

8. SMTP error messages

If a message is rejected because of RMX records, an error message should be issued which explains the details. It is to be discussed whether new SMTP error codes are to be defined. Error messages should be verbose to make debugging of configuration errors easy.

9. Message relaying and forwarding

9.1. Problem description

Message forwarding and relaying means that an MTA which received an e-mail by SMTP does not deliver it locally, but resends the message - usually unchanged except for an additional Received header line and maybe the recipient's address rewritten - to another SMTP MTA. Message forwarding is an essential functionality of e-mail transport services, for example:

- Message transport from outer MX relay to the intranet
- Message forwarding and Cc-ing by .forward or .procmail-alike mechanisms
- Mailing list processing
- Message reception by mail relays with low MX priority, usually provided by third parties as a stand-by service in case of relay failure or maintenance
- "Forwarding" and "Bouncing" as a MUA functionality

In all of these cases a message is sent by SMTP from a host which is not covered by the original sender domain's RMX records. While the RMX records would forbid accepting this message, it still must be accepted. The following subsections explain how to cope with relaying.

9.2. Trusted relaying/forwarding

In some cases the receiving MTA trusts the sending MTA to not fake messages and to already have checked the RMX records at message reception. As a typical example, a company might have an outer mail relay which receives messages from the Internet and checks the RMX records. This relay then forwards the messages to the several department's mail servers. It does not make sense for these department mail servers to check the RMX records, because the RMX records have already been checked and because they would always reject messages, since the relay is not covered by the originator's RMX records. In this case there is a trust relationship between the department relays and the outer relay. So RMX checking is turned off for trusted relays. In this example, the department relays would not check messages from the outer relay (but for intranet security, they could still check RMX records of the other

departments sub-domains to avoid internal forgery between departments).

Another common example are the low-priority MX relays, which receive and cache e-mails when the high-priority relays are down. In this case, the high-priority relay would trust the low-priority

relay to have verified the sender authorization and would not perform another RMX verification (which would obviously fail).

When a relay forwards a message to a trusting machine, the envelope sender address should remain unchanged.

9.3. Untrusted relaying/forwarding

If the receiving MTA does not trust the forwarding MTA, then there is no chance to leave the sender envelope address unchanged. At a first glance this might appear impracticable, but this is absolutely necessary. If an untrusted MTA could claim to have forwarded a message from a foreign sender address, it could have forged the message as well. Spammers and forgers would just have to act as such a relay.

Therefore, it is required that, when performing untrusted forwarding, the envelope sender address has to be replaced by the sender address of someone responsible for the relaying mechanism, e.g. the owner of the mailing list or the mail address of the user who's forwarding mechanism caused the transmission. It is important to stress that untrusted relaying/forwarding means taking over responsibility for the message. It is the idea of RMX records to tie responsibility to message transmission. Untrusted relaying without replacing the sender address would mean to transmit without taking responsibility.

The disadvantage is that the original sender address is lost. Therefore, whenever a sender address replacement happens, the Received-Line must contain the old address. Many of today's MTAs already insert the envelope recipient address, but not the sender address into the Received header line. It seems reasonable to require every Received line to include both the sender and recipient address of the incoming SMTP connection.

10. Further development and improvements of RMX

This RFC is intended to be close to the earlier RMX drafts for historical reasons. Therefore, the further development and improvements are not made "in place" but described in this new chapter.

10.1. Separate RMX records for address types

In earlier draft versions of this memo there was only one RMX record covering all possible identity classes provided by the authentication step, e.g. IPv4 and IPv6 addresses were described by the same RMX record.

This does not make sense and unnecessarily inflates the RMX records. Since the receiving MTA knows the identity class at query time, separate RMX records can be provided for each supported identity class, e.g. different RMX records for IPv4 and IPv6 addresses.

So RMX records could look like

```
_ipv4._rmx.danisch.de   IN RMX ipv4:213.133.101.23
_ipv6._rmx.danisch.de   IN RMX ipv6:fec0::0/16
```

10.2. SCAF - Simple Caller Authorization Framework

Fraud, spam, worms, spoofing are not limited to SMTP only. Other internet protocols like news transfer, chat and instant messaging, and even non-internet protocols can be protected against spoofing with RMX. It could also become a simple, password-less caller identification mechanism for protocols like HTTP or FTP. For example, a web browser could provide a user address similar to an e-mail address as a HTTP [7] cookie, in a new request header entry type, or as a password-less HTTP authentication header.

Imagine there is a vendor's web server providing web pages available for employees of a particular company only (e.g. software upgrades for a customer). The customer could be required to provide an RMX-like record describing it's network which allows the

web server to limit access based on the given user address and the RMX record for this application. (It can be understood as a kind of external firewall rule. Even firewalls and proxies could support it.)

To distinguish records for different applications, the records must be stored at different locations in DNS, e.g.

```
_ipv4._smtp._rmx.danisch.de   IN RMX ipv4:213.133.101.23
_ipv4._http._rmx.danisch.de   IN RMX host:homeoffice.danisch.de
```

would restrict access to web servers with "hadmut@danisch.de" to the IP address pointed to by the A record of homeoffice.danisch.de (which can be a dynamically assigned address).

10.3. RMX++

RMX has restrictions and might not be applicable or desirable in all cases due to it's inflexible record types, it's nature to reveal the domain's network structure, and limitations of DNS. The domain owner would always be limited to those entry types commonly defined.

To overcome these restrictions, the successor RMX++ significantly differs from RMX. With RMX++, the query is split into two distinct steps. In a first step, the receiving MTA takes the domain name of the sender address to query DNS as with RMX. But instead of an RMX record, the MTA queries an A record, an SRV record [8], and a TXT record.

The A or SRV records are then used to locate a server for the second step. The TXT record contains a URL pattern. A default pattern is used in absence of the TXT record. The pattern contains macros which are to be expanded, e.g. substituted with the server address, the mail sender address, the calling MTA's IP address, message ID, content type etc. The MTA then fetches the RMX record found at that URL. The preferred protocol type is HTTP or HTTPS, but other protocols could be used as well. Even DNS and LDAP queries can be described in URLs.

In general, there are three types of RMX records: In the first case, the RMX record is a static one and stored as a file on the web server. In the second case, it is dynamically generated by the web server (e.g. through a CGI program). In the third case, the MTA passes required information to the server and the server replies with "Allow" or "Reject". The receiving MTA does query and handle all three types the very same way.

This method has several advantages:

- No need for a new DNS RR type. DNS servers don't need to be upgraded.

- Easier to administer: Today, every domain owner is able to put a file on a web site. Helper programs run on the web server will help to generate the RMX records with

easy and foolproof user interfaces. In contrast, DNS zone tables are more difficult and not as easy available for modification for everyone at any time.

- There is no size limit for the record as with DNS. The RMX record does not need to be split into several DNS records and stitched together. The encoding does not need to be as tight as described above, and can be plain text, ASN.1, XML, etc.

- Simple encryption with HTTPS

- Support of full sender address verification (not just the domain part) is trivial.

- Caching with HTTP proxies, caching control and expiry with HTTP headers

- The RMX record can be generated dynamically on request. Large sites where thousands of users log in and out all the time (e.g. large mail service providers) can provide "fresh" records for every request without the need to update their zone table every second.

  Dynamic RMX records can be easily generated with CGI applications, a well known and robust mechanism.

- RMX records can be smaller because they need to cover only the query sent to the server and don't need to describe the full network structure which might consist of thousands of computers.

- RMX verification can be moved from the receiving MTA to the domain owner's server, because the MTA can pass all required data such as sender address, IP addresses, size, message ID, etc. as CGI parameters. It allows the domain owner to completely hide his network structure and the authorization method, and to implement any arbitrary mechanism. Just as an extreme example to point out the capability, the domain owner's server could calculate a

horoscope on request and decide whether to permit or not based on whether the planet constellation promises the e-mail to be lucky. Obviously, in reality the domain owner would use some kind of database to verify the sender. The domain owner is free to implement anything.

- If the domain owner chooses to use a URL with CGI parameters, to use HTTPS, or to instruct caches to not cache, the server

will be queried for every single e-mail. This allows the server to limit the number of e-mails sent and to detect anomalies, e.g. that a machine has been infected by some worm or virus.

11. Security Considerations

11.1. Draft specific considerations

11.1.1. Authentication strength

It is important to stress, that the suggested method does not provide high level security and does not completely prevent forged e-mails or spam under any circumstances. It is a robust, but not highly reliable and completely secure security mechanism. Keep in mind that it is based on DNS, and DNS is not secure today. Authorization is based on the IP address. The very same machine with the very same IP address could be authorized to send e-mail with a given sender address and sending spam at the same time. Maybe because several users are logged in. Or because several customers use the same relay of the same ISP, where one customer could use the sender address of a different customer. It is up to the ISP to prevent this or not. Machines can still be hijacked. Spammers are also domain owners. They can simply use their own domain and authorize themselves. You will always find people on the world who do not care about security and open their relays and RMX records for others to abuse them. RMX is to be considered as a very cheap and simple light weight mechanism, which can nevertheless provide a significant improvement in mail security against a certain class of attacks, until a successor of SMTP has been defined and commonly accepted.

11.1.2. Where Authentication and Authorization end

Early versions of RMX drafts did not cover the local part of the e-mail address, i.e. what's on the left side of the @ sign. This is still to be discussed. Authentication and authorization are limited to the sending MTA's IP address. The authentication is limited to the TCP functionality, which is sufficient for light weight authentication. The RMX records authorize the IP address of the sending host only, not the particular sender of the message. So if a machine is authorized to use sender addresses of more than a single domain, the authentication scheme does not prevent that any user on this machine can send with any of these domains. RMX is not a substitute for the host security of the involved machines.

The proposed authentication scheme can be seen as a "half way

authentication": It does not track back an e-mail to the effective sender. It tracks only half of the way, i. e. it tracks back to the domain and it's DNS administrators who authorized that particular sender IP address to use it for sending e-mail. How the party responsible for that domain performs user authentication, whom it grants access to, how it helds people responsible for

abuse, is completely left as the private business of those who are in charge of that domain. So this draft does not interfere with the domain's individual security policy or any legislation about such policies. On the other hand, the proposed authentication scheme does not give any statement about the nature and quality of the domain's security policy. This is an essential feature of the proposal: E-mail authentication must be deployed world wide, otherwise it won't do the job. Any security scheme interfering with the local legislations or the domain's security policy will not be accepted and can't effectively deployed. Therefore, the security policy must remain the domain's private business, no matter how lousy the policy might be.

In order to achieve this and to make use of the only existing world wide Internet directory scheme (DNS), the approach of this proposal is to just ignore the local part of the sender address (i.e. what's left of the @ part) and limit view to the domain part. After all, that's what we do anyway when delivering to a given address with SMTP.

11.1.3. Vulnerability of DNS

DNS is an essential part of the proposed authentication scheme, since it requires any directory service, and DNS is currently the only one available. Unfortunately, DNS is vulnerable and can be spoofed and poisoned. This flaw is commonly known and weakens many network services, but for reasons beyond that draft DNS has not been significantly improved yet. Several commentors to previous drafts asked to not use DNS because of its lack of security. This is unfeasible: Any authentication/authorization system linked to some kind of symbolic identity (in this case the domain name) needs some kind of infrastructure and trusted assignment. There are basically two ways to do it: Do it yourself and trust nobody else, or let someone else do it. There are methods to do it the former way, e.g. to give someone some kind of authentication/authorization information after a first successful e-mail exchange, e.g. some kind of cookie or special e-mail address. This is certainly interesting and powerful, but it does not solve the problem on a world wide scale and is far to complicated and error prone for the average user, i. e. 99% of the users.

The latter method to let someone else do the symbolic name assignment and create the authentication framework is well known. It context of public key cryptography, this is called a Public Key

Infrastructure (PKI). One of the best known facts about PKIs is that, until now, we don't have any covering a significant part of the Internet. And we won't have any in near future. The complexity is far too high, it is too expensive, and it involves

cooperation of every single user, which is simply unrealistic and extremely error prone. So what do we have we can use? All we have is the DNS and the Whois database. And we have countries who don't allow cryptography. So the proposal was designed to use DNS without cryptography. It does not avoid DNS because of its vulnerability, it asks for a better DNS, but accepts the DNS as it is for the moment. Currently there are two main threats caused by the DNS weakness:

- A spammer/forger could spoof DNS in order to gain false authorization to send fake e-mails.

- An attacker could spoof DNS in order to block delivery from authorized machines, i. e. perform a Denial of Service attack.

The first one is rather unrealistic, because it would require an average spammer to poison a significant part of the DNS servers of its victims. A spammer sending messages to one million receipients would need to poison at least 1-10% which is 10,000 to 100,000 receipient's DNS servers. This should be unfeasible in most cases.

In contrast, the second threat is a severe one. If an attacker wanted to block messages from one company to another, he just needs to poison the recipients DNS server with a wrong RMX record in order to make the recipient's SMTP machine reject all messages. And this is feasible since the attacker needs to poison only a single DNS server. But does this make SMTP more vulnerable? No. Because the attacker can already do even more without RMX. By poisoning the sender's DNS server with wrong MX records, the attacker can also block message delivery or even redirect the messages to the attacker's machine, thus preventing any delivery error messages and furthermore getting access to the messages.

As a consequence, e-mail delivery by SMTP requires a better DNS anyway. The requirements are not significantly expanded by RMX.

11.1.4. Sneaking RMX attack?

A certain kind of sneaking DNS attack could be possible. DNS and RMX implementors should take care to void it.

Imagine an unauthorized sender is sending a forged mail (e.g. spam). At connection time, before querying the RMX record, the receiving MTA usually performs a PTR query for the IP address of the sending MTA. If the sender has control over the authoritative name server for that particular IP address, the sender could give a normal PTR answer, but could append a wrong RMX, APL, or A record in the additional section of the query. A subsequent RMX query

could receive wrong DNS data if the DNS server used by the receiving MTA accepted those forged records.

11.1.5. Open SMTP relays

Open SMTP relays (i.e. machines which accept any e-mail message from anyone and deliver to the world) abused by spammers are a one of the main problems of spam defense and sender backtracking. In most cases this problem just vanishes because foreign open relay machines will not be covered by the RMX records of the forged sender address. But there are two special cases:

If the spammer knows about a domain which authorizes this particular machine, that domain can be abused for forgery. But in this case, the IP address of the relay machine and the RMX records of the domain track back to the persons responsible. Both can be demanded to fix the relay or remove the RMX record for this machine. An open relay is a security flaw like leaving the machine open for everybody to login and send random mails from inside. Once the administrative persons refuse to solve the problem, they can be identified as spammers and held responsible.

The second special case is when a domain authorizes all IP addresses by having the network 0.0.0.0/0 in the RMX/APL record. In this case, open relays don't make things worse. It's up to the recipient's MTA to reject mails from domains with loose security policies.

11.1.6. Unforged Spam

RMX does not prevent spam (which is, by the way, not yet exactly defined), it prevents forgery. Since spam is against law and violates the recipients rights, spam depends on untracability of the sender. In practice the sender forges the sender address (other cases see below). RMX is designed to detect such forgeries.

However, the RMX approach is rendered ineffective, if the sender does not forge. If the sender uses just a normal address of his own domain, this is just a plain, normal e-mail, which needs to be let through. Since it is up to the human's taste whether this is spam or not, there's no technical way to reliably identify this as

spam. But since the sender domain is known, this domain can be blacklisted or legal steps can be gone into.

11.1.7. Reliability of Whois Entries

Once the RMX infrastructure gets deployed, what's the security gain? It allows to determine the domain which's DNS zone

authorized the sending machine. What's that good for? There are some immediate uses of the domain name, e.g. in black- and whitelisting. But in most cases this is just the starting point of further investigations, either performed automatically before message acceptance, or manually after spam has been received and complainted about.

The next step after determining the domain is determining the people responsible for this domain. This can sometimes be achieved by querying the Whois databases. Unfortunately, many whois entries are useless because they are incomplete, wrong, obsolete, or in uncommon languages. Furthermore, there are several formats of address informations which make it difficult to automatically extract the address. Sometimes the whois entry identifies the provider and not the owner of the domain. Whois servers are not built for high availability and sometimes unreachable.

Therefore, a mandatory standard is required about the contents and the format of whois entries, and the availability of the servers. After receiving the MAIL FROM SMTP command with the sender envelope address, the receiving MTA could check the RMX record and Whois entry. If it doesn't point to a real human, the message could be rejected and an error message like "Ask your provider to fix your Whois entry" could be issued. Obviously, domain providers must be held responsible for wrong entries. It might still be acceptable to allow anonymous domains, i. e. domains which don't point to a responsible human. But it is the receivers choice to accept e-mails from such domains or not.

11.1.8. Hazards for Freedom of Speech

Currently, some governments try to enforce limitations of internet traffic in order to cut unwanted content providers from the network. Some of these governments try to hide a whole country behind firewalls, others try to force Internet providers to poison DNS servers with wrong A records for web servers, e.g. one county administration in Germany tries to do so. If message reception depends on DNS entries, the same governments will try to block not only HTTP, but SMTP from these domains also.

However, since most MTAs already reject messages from unresolvable domain names this is not a new threat.

11.2. General Considerations about spam defense

After discussing security requirements of the proposal, now the security advantages of the RMX approach over content based filters will be explained. Basically, there are three kinds of content

filters:

- Those which upload the message or some digest to an external third party and ask "Is this spam"?

- Those which download a set of patterns and rules from a third party and apply this set to incoming messages in order to determine whether it is spam.

- Those which are independent and don't contact any third party, but try to learn themselves what is spam and what isn't.

The message filters provided by some e-mail service providers are usually not a kind of their own, but a combination of the first two kinds.

11.2.1. Action vs. reaction

Content filters suffer from a fundamental design problem: They are late. They need to see some content of the same kind before in order to learn and to block further distribution.

This works for viruses and worms, which redistribute. This doesn't work for spam, since spam is usually not redistributed after the first delivery. When the filters have learned or downloaded new pattern sets, it's too late.

RMX does not have this problem.

11.2.2. Content based Denial of Service attacks

All three kinds of content filters, but especially the second and the third kind are vulnerable to content based Denial of Service attacks.

If some kind of third party (e.g. non-democratic government, intellectual property warriors, religious groups, military, secret

services, patriots, public relation agents, etc.) wants certain contents not to be distributed, they could either poison the pattern/rule databases or feed wrong sets to particular receivers.

Such pattern/rule sets are the perfect tool for censoring e-mail traffic and denial of service attacks by governments and other parties, and a similar threat are virus filters. E. g. the content industry could demand to teach all virus and spam filters to delete all e-mails containing the URL of an MP3 web server outside the legislation. Software manufacturers could try to block all e-mails

containing software license keys, thus trying to make unallowed distribution more difficult. Governments could try to block distribution of unwanted information and politically incorrect speech.

RMX does not have this problem.

12. Privacy Considerations

(It was proposed on the 56th IETF meeting to have a privacy section in drafts and RFCs.)

12.1. Draft specific considerations

12.1.1. No content leaking

Since the RMX approach doesn't touch the contents of a message in any way, there is obviously no way of leaking out any information about the content of the message. RMX is based solely on the envelope recipient address. However, methods to fix problems not covered by RMX might allow content leaking, e.g. if the acceptance of a message with an empty sender address requires the reference to the message id of an e-mail recently sent, this allows an attacker to verify whether a certain message was delivered from there.

12.1.2. Message reception and sender domain

Message delivery triggers RMX and APL requests by the recipient. Thus, the admin of the DNS server or an eavesdropper could learn that the given machine has just received a message with a sender from this address, even if the SMTP traffic itself had been encrypted.

However, most of today's MTAs do query the MX and A records of the domain after the MAIL FROM command, so this is not a real new threat.

12.1.3. Network structure

Since RMX and its associated APL records provide a complete list of all IP addresses of hosts authorized to send messages from this address, they do reveal informations about the network structure and maybe the lifestyle of the domain owner, since a growing number of domains are owned by single persons or families. E.g. the RMX records could reveal where someone has his job or spends his time at weekends.

If such informations are to be kept secret, it is the user's job to not sent e-mails from there and to relay them from non-compromising IP addresses.

12.1.4. Owner information distribution

As described above, RMX depends partly on the reliability of the whois database entries. It does not make anonymous domains

impossible, but it requires to keep the database entries "true", i. e. if a whois entry does not contain informations about the responsible person, this must be unambigously labeled as anonymous. It must not contain fake names and addresses to pretend a non-existing person. However, since most Internet users on the world feel extremely annoyed by spam, they will urge their MTA admin to reject messages from anonymous domains. The domain owner will have the choice to either remain anonymous but be not able to send e-mail to everyone in the world, or to be able but to reveal his identity to everyone on the world.

It would be possible to provide whois-like services only to recipients of recent messages, but this would make things too complicated to be commonly adopted.

12.2. General Considerations about spam defense

12.2.1. Content leaking of content filters

As described above in the Security chapter, there are spam filters which inherently allow leakage of the message body. Those filters upload either the message body, or in most cases just some kind of checksum to a third party, which replies whether this is to be seen as spam or not. The idea is to keep a databases of all digests of all messages. If a message is sent more often than some threshold, it is to be considered as a mass mail and therefore tagged as spam.

While the digest itself does not reveal the content of the message, it perfectly reveals where a particular message has been delivered to. If a government finds just a single unwanted message, if a software manufacturer finds a single message with a stolen product license key, if someone finds a message with unpatriotic content, it takes just a single database lookup to get a list of all people who received this particular message. Content filters with digest upload are "Big Brother's" favourite toy.

12.2.2. Black- and Whitelists

Some proposals against spam are based on a central database of white- or blacklisted IP addresses, Sender names, Message IDs or whatever. Again, there is a central database which learns who has

received which e-mail or from which sender with every query. This allows tracking relations between persons, which is also a breach of privacy.

13. Deployment Considerations

13.1. Compatibility

13.1.1. Compatibility with old mail receivers

Since the suggested extension doesn't change the SMTP protocol at all, it is fully compatible with old mail receivers. They simply don't ask for the RMX records and don't perform the check.

13.1.2. Compatibility with old mail senders

Since the SMTP protocol is unchanged and the SMTP sender is not involved in the check, the method is fully compatible with old mail senders.

13.1.3. Compatibility with old DNS clients

Since the RMX is a new RR, the existing DNS protocol and zone informations remain completely untouched.

If RMX is provided as a TXT record instead, it must be ensured that no other software is misinterpreting this entry.

13.1.4. Compatibility with old DNS servers

Full compatibility: If the server does not support RMX records, RMX in TXT records can be used.

13.2. Enforcement policy

Obviously, for reasons of backward compatibility and smooth introduction of this scheme, RMX records can't be required immediately. Domains without RMX records must temporarily be treated the same way as they are treated right now, i.e. e-mail must be accepted from anywhere. But once the scheme becomes sufficiently widespread, mail relays can start to refuse e-mails

with sender addresses from domains without RMX records, thus forcing the owner of the domain to include a statement of authorization into the domain's zone table. Domain owners will still be free to have an RMX record with a network and mask 0.0.0.0/0, i.e. to allow e-mails with that domain from everywhere. On the other hand, mail receivers will be free to refuse mails from domains without RMX records or RMX records which are too loose. Advanced MTAs might have a configuration option to set the maximum number of IP addresses authorized to use a domain. E-mails from a domain, which's RMX records exceed this limit, would be rejected. For example, a relay could reject e-mails from domains which

authorize more than 8 IP addresses. That allows to accept e-mails only from domains with a reasonable security policy.

14. General considerations about fighting spam

Is there a concise technical solution against spam? Yes.

Will it be deployed? Probably not.

Why not? Because of the strong non-technical interests of several parties against a solution to the problem, as described below. Since these are non-technical reasons, they might be beyond the scope of such a draft. But since they are the main problems that prevent fighting spam, it is unavoidable to address them.

14.1. The economical problem

As has been recently illustrated in the initial session of the IRTF's Anti Spam Research Group (ASRG) on the 56th IETF meeting, sending spam is a business with significant revenues.

But a much bigger business is selling anti-spam software. This is a billion dollar market, and it is rapidly growing. Any simple and effective solution against spam would defeat revenues and drive several companies into bankrupt, would make consultants jobless.

Therefore, spam is essential for the anti-spam business. If there is no spam, then no Anti-Spam software can be sold, similar to the anti-virus business. There are extremely strong efforts to keep this market growing. Viruses, Worms, and now spam are just perfect to keep this market alive: It is not sufficient to just buy a software. Databases need to be updated continuously, thus making the cash flow continuously. Have a single, simple, and permanent solution to the problem and - boom - this billion dollar market is dead. That's one of the reasons why people are expected to live with spam. They have to live with it to make them buy anti-spam software. Content filters are perfect products to keep this market alive.

14.2. The POP problem

Another problem is the history of mail delivery. Once upon a time,

there used to be very few SMTP relays which handled the e-mail traffic of all the world, and everybody was happy with that. Then odd things like Personal Computers, which are sometimes switched off, portable computers, dynamicaly assigned IP addresses, IP access from hotel rooms, etc. was invented, and people became unhappy, because SMTP does not support delivery to such machines. To make them happy again, the Post Office Protocol[9] was invented, which turned the last part of message delivery from SMTP's push

style into a pull style, thus making virtually every computer on the world with any random IP address a potential receiver of mails for random domains. Unfortunately, only receiving e-mail was covered, but sending e-mail was left to SMTP.

The result is that today we have only very few SMTP relays pointed to by MX records, but an extreme number of hosts sending e-mail with SMTP from any IP address with sender addresses from any domain. Mail delivery has become very asymmetric. Insecurity, especially forgeability, has become an essential part of mail transport.

That problem could easily be fixed: Use protocols which allow uploading of messages to be delivered. If a host doesn't receive messages by SMTP, it shouldn't deliver by SMTP. Mail delivery should go the same way back that incoming mail went in. This is not a limitation to those people on the road who plug their portable computer in any hotel room's phone plug and use any provider. If there is a POP server granting download access from anywhere, then the same server should be ready to accept uploading of outgoing messages.

But as comments on the first draft version of this RFC showed, people religiously insist on sending e-mail with their domain from any computer with any IP address in the world, e.g. when visiting a friend using her computer. It appears to be impossible to convince people that stopping mail forgery requires every one of them to give up forging.

14.3. The network structure problem

A subsequent problem is that many organisations failed to implement a proper mail delivery structure and heavily based their network on this asymmetry. The author received harsh comments from Universities who were unable to give their network a reasonable structure. While they do have a central mail relay for incoming mail to the universities domain, they developed a structure where every member of the University randomly sends e-mails with that University's domain as a sender address from home or everywhere in the world with any dynamically assigned IP address from any provider. So this domain is to be used from every possible IP address on earth, and they are unable to operate any authentication scheme. Furthermore, they were unable to understand that such a

policy heavily supports spam and that they have to expect that people don't accept such e-mails anymore once they become blacklisted.

As long as organisations insist on having such policies, spammers

will have a perfect playground.

14.4. The mentality problem

Another problem is the mentality of many internet users of certain countries. The author received harsh comments from people who strongly insisted on the freedom to send any e-mail with any sender address from anywhere, and who heavily refused any kind of authentication step or any limitation, because they claimed that this would infringe their constitutional "Freedom of Speech". They are undeviatingly convinced that "Freedom of Speech" guarantees their right to talk to everybody with any sender address, and that is has to be kept the recipient's own problem to sort out what he doesn't want to read - on the recipient's expense. The author learned that it is extremely difficult to convince some people to give up random e-mail sending. However, a security mechanism for the world wide mail system can never meet the taste and the requirements of every single one of all those hundreds of millions of users.

It requires a clear statement that the constitutional "Freedom of Speech" does not cover molesting people with unsolicited e-mail with forged sender address.

14.5. The identity problem

How does one fight against mail forgery? With authentication. What is authentication? In simple words: Making sure that the sender's real identity meets the recipients idea of who is the sender, based on the sender address which came with the message.

What is identity? It is the main problem. Several countries have different ideas of "identity", which turn out to be somehow incompatible. In some countries people have identity cards and never change their name and birthday. Identities are created by human birth, not by identity changes. Other countries do not have such a tight idea about identity. People's temporary identity is based on nothing more than a driving license and a social security number. With this background, it is virtually impossible to create a trustworthy PKI covering all Internet users.

14.6. The multi-legislation problem

Many proposals about fighting spam are feasible under certain legislations only, and are inacceptable under some of the legislations. But a world wide applicable method is required. That's why the approach to ask everone on the world to sign messages with cryptographic keys is not feasible.

References

1. **J. Klensin, "Simple Mail Transfer Protocol," RFC 2821 (April 2001).**

2. **P. Resnick, "Internet Message Format," RFC 2822 (April 2001).**

3. **P. Koch, "A DNS RR Type for Lists of Address Prefixes (APL RR),"** RFC 3123 (June 2001).

4. **T. Dierks, C. Allen, "The TLS Protocol," RFC 2246 (January 1999).**

5. **J. Myers, "Simple Authentication and Security Layer (SASL),"** RFC 2222 (October 1997).

6. **P. Mockapetris, "DOMAIN NAMES - IMPLEMENTATION AND SPECIFICATION,"** RFC 1035 (November 1987).

7. **T. Berners-Lee and others, "Hypertext Transfer Protocol HTTP/1.1,"** RFC 2616 (June 1999).

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

9. **J. Myers, M. Rose, "Post Office Protocol - Version 3," RFC 1939** (May 1996).

History

```
1992-... Research on organisational E-Mail security
Dec 2002 Internet-Draft 00
Mar 2003 IRTF's Anti Spam Research Group started it's work
Apr 2003 Internet-Draft 01
Jun 2003 Internet-Draft 02
Oct 2003 Internet-Draft 03
```

Author's Address

Hadmut Danisch	rfc@danisch.de
Tennesseeallee 58	http://www.danisch.de
76149 Karlsruhe	Phone: +49-721-843004
Germany	Phone: +49-351-4850477

Comments

Please send comments to rfc@danisch.de.