Internet Draft
Category: Experimental
draft-mengwong-spf-01.txt
Expires: September 2004

Mark Lentczner
Meng Weng Wong, pobox.com

May 2004

Sender Policy Framework (SPF)
A Convention to Describe Hosts Authorized to Send SMTP Traffic

Status of this Memo

Abstract

Email address forgery is a problem on the Internet today. Domain owners want to control the use of their names in email, but are helpless because they lack the means. This document introduces a language for domains to make email-related declarations in DNS. It defines in detail one possible sender authentication scheme for domains to describe the hosts from which they send mail. SMTP receivers can use this scheme to detect sender forgery.

Table of Contents

1. Introduction

The intended audience for this document includes administrators of the Domain Name System and developers of Mail Transfer Agents (MTAs), Mail Delivery Agents (MDAs), and Mail User Agents (MUAs). They are assumed to be familiar with the workings of SMTP and DNS. See [RFC2821] and [RFC1034].

Forgery of domain names is a problem. Malicious entities often falsify envelope and header addresses to make it harder to identify the source of a message. When those addresses correspond to real people and organizations, the victims of forgery suffer a cost in bounce messages, tarnished reputations, and misdirected abuse reports.

SPF is designed to fight email address forgery. It does this by establishing a policy framework and an authentication scheme.

SPF defines a simple language. Domains can use that language to describe the mail they send. SMTP receivers can use these descriptions to evaluate messages.

SPF can implement a sender authentication scheme. In a sender authentication scheme, a domain owner asserts that legitimate messages from that domain must meet certain criteria. Messages which do not meet the criteria are not legitimate. These assertions are made in machine-readable form.

SPF defines a specific sender authentication scheme based on the designated sender model. A domain identifies certain hosts as

designated senders. Mail from those hosts is considered legitimate. Mail from other hosts is not.

SPF publishes policy data in the DNS. DNS resolvers can cache SPF data. Caching reduces lookup traffic. Sender domains do not have to run new servers to advertise SPF information.

SPF is extensible. Multiple mechanisms can be defined. While other sender authentication schemes can be expressed in SPF, the rest of this document defines a designated sender scheme in detail.

1.1 Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

An SMTP client sends mail to an SMTP server. The SMTP server and downstream systems comprise the SMTP receiver. The SMTP receiver acts as an SPF client to an SPF publisher domain.

SPF processing may occur as early as the MAIL FROM stage of an SMTP transaction or as late as the display stage in a Mail-User Agent. For convenience, SMTP servers which accept, classify, discard, or reject mail on the basis of SPF tests may be said to be speaking "SMTP+SPF".

1.2 Designated Senders

Designated sender schemes are weaker than cryptographic schemes but provide more assurance than the current SMTP model.

SPF defines a set of mechanisms which add up to a designated sender scheme. First, domain owners designate legitimate outbound mail servers in a compact, symbolic notation. SMTP receivers may query sender domains using these mechanisms and decide the validity of a given SMTP transaction while that transaction is ongoing, even before any message data is passed. Alternatively, SPF tests can be performed after SMTP time by an MDA or MUA. MTAs, MDAs, and MUAs may choose to accept, classify, discard, or reject messages based on the result of an SPF test.

SPF can be used to verify the sender of a message based on envelope information available at SMTP time or according to the headers after an SMTP transaction has completed.

2. SPF Records

Domains declare verifiable attributes that describe the mail they send.

A domain's declarations are presented in an SPF record. The record is a single string of text:

```
SPF-record  = version *( 1*SP ( directive | modifier ) )
```

An example SPF Record is:

```
v=spf1 +mx +a:colo.example.com/28 -all
```

This record has a version of "v=spf1" and three directives: "+mx", "+a:colo.example.com/28", and "-all".

2.1 Publishing

The SPF record is published in the DNS. The record is of type TXT. The record is placed in the DNS tree at the level of the domain.

The TXT type was chosen for pragmatic reasons.

SPF clients ignore records which do not carry a recognized version string. This document specifies the version string "v=spf1".

A domain MUST NOT return multiple records that begin with the version "v=spf1". If more than one "v=spf1" record is returned, this constitutes a syntax error and the result is "unknown".

Note: The comparison is done on the entire version section (which is terminated either by a SP character, or the end of the TXT record). Hence, a record with a version of "v=spf10" is not considered a record with version "v=spf1".

An example SPF record is:

```
v=spf1 +mx +a:colo.example.com/28 -all
```

This might be published easily via this line in a domain file:

```
example.com. IN TXT "v=spf1 +mx +a:colo.example.com/28 -all"
```

In unusual situations, directives may require additional DNS records. If additional records are used, they MAY be published under the "_spf" subdomain. See Appendix B for examples.

An SPF record MAY consist of a single TXT record with multiple strings. If such an TXT record is encountered, then an SPF client MUST concatenate those strings without adding spaces, eg

```
TXT "v=spf1 .... first" "second string..."
```

MUST be treated as equivalent to

```
TXT "v=spf1 .... firstsecond string..."
```

TXT records containing multiple strings are useful in order to construct more complex SPF records which would otherwise exceed the maximum length of a string within a TXT record.

Note: Many nameserver implementations will silently split long strings in TXT records into several shorter strings.

2.2 Interpretation

SPF clients are applications that parse and interpret the SPF record for a domain. Clients MUST correctly interpret the SPF record according to the canonical algorithm defined here.

Clients MAY use a different algorithm, so long as the results are the same.

2.2.1 Subject of SPF testing

In an SMTP transaction, an SMTP client may provide FQDNs in the HELO argument and in the MAIL FROM return-path. SMTP+SPF receivers MAY check the HELO argument and MUST check the return-path. A single SMTP transaction may therefore trigger one or two SPF queries.

Accordingly, the <responsible-sender> may be drawn from the HELO argument or from the "MAIL FROM" return-path. This document sometimes refers to the <responsible-sender> as the "envelope sender".

It is RECOMMENDED that SMTP+SPF receivers perform tests using the following algorithm.

SMTP+SPF receivers MAY check the HELO argument. In this mode, the <responsible-sender> comes from the HELO argument IF the HELO argument is a fully qualified domain name. If the HELO argument is not an FQDN, there is nothing to check and the result is "unknown". If the HELO test returns a "fail", the overall result for the envelope is "fail", and there is no need to test the return-path.

SMTP+SPF receivers MUST check the return-path unless HELO testing produced a "fail". In this mode, the <responsible-sender> comes from the domain name of the "MAIL FROM" return-path. When the return-path has no domain, a client MUST use the HELO domain instead. If the HELO argument does not provide an FQDN, SPF processing terminates with "unknown".

If SPF processing occurs after SMTP time, the envelope sender may be obtained from the Return-Path header. If the Return-Path header has no domain, a client MAY try to extract the HELO domain from the Received headers. If the headers do not yield useful envelope information, SPF processing terminates with "unknown".

SMTP+SPF receivers MAY test the domain given in the HELO argument whether or not the return-path contains a domain name.

However, the <responsible-sender> address MAY be drawn from an alternative source. For example, an MUA may find it more convenient to extract the <responsible-sender> from the Return-Path header or from the Sender: header.

If the <responsible-sender> has no localpart, clients MUST substitute the string "postmaster" for the localpart.

The <current-domain> is initially drawn from the <responsible-sender>. Recursive mechanisms such as Include and Redirect replace the initial <current-domain> with another domain. However, they do not change the value of the <responsible-sender>. See sections 4.2, 3.3, and 8.4.

2.2.2 Lookup

SPF clients perform a TXT type query in search of an SPF record.

Any number of records may be returned. Only the record which begins with the version "v=spf1" is relevant to this document. CNAME responses are followed as usual.

If no matching records are returned, an SPF client MUST assume that the domain makes no SPF declarations. SPF processing MUST abort and return "none".

If the domain does not exist (NXDOMAIN), SPF clients MUST return "unknown".

If a domain has no SPF record, clients MAY substitute SPF data from a parent domain ONLY IF the appropriate parent domain's SPF record sets "match_subdomains=yes". For example, if no SPF record is

found for "workstation.example.com", clients MAY proceed to automatically query "example.com". The appropriate parent domain to fallback on MUST be determined according to the DNS zone cut.

3. SPF Record Evaluation

An SPF client evaluates an SPF record and produces one of seven results:

None: The domain does not publish SPF data.

Neutral (?): The SPF client MUST proceed as if a domain did not publish SPF data. This result occurs if the domain explicitly specifies a "?" value, or if processing "falls off the end" of the SPF record.

Pass (+): the message meets the publishing domain's definition of legitimacy. MTAs proceed to apply local policy and MAY accept or reject the message accordingly.

Fail (-): the message does not meet a domain's definition of legitimacy. MTAs MAY reject the message using a permanent failure reply code. (Code 550 is RECOMMENDED. See [RFC2821] section 7.1.)

Softfail (~): the message does not meet a domain's strict definition of legitimacy, but the domain cannot confidently state that the message is a forgery. MTAs SHOULD accept the message but MAY subject it to a higher transaction cost, deeper scrutiny, or an unfavourable score.

There are two error conditions, one temporary and one permanent.

Error: indicates an error during lookup; an MTA SHOULD reject the message using a transient failure code, such as 450.

Unknown: indicates incomplete processing: an MTA MUST proceed as if a domain did not publish SPF data.

When SPF-aware SMTP receivers accept a message, they SHOULD prepend a Received-SPF header. See section 6.

SPF clients MUST use the algorithm described in this section or its functional equivalent.

If an SPF client encounters a syntax error in an SPF record, it must terminate processing and return a result of "unknown".

3.1 Matching Version

An SPF record begins with a version section:

```
version = "v=spf" version-number
version-number = 1*DIGIT
```

SPF clients MUST use only the records of the highest understood version published by a domain and ignore all lower versions, unless that version explicitly recognizes lower versioned responses.

For example, if an SPF client understands versions 1, 2 and 3, and the DNS query results in records of version 1, 2 and 4, then only the record with version 2 is used.

This specification describes version 1. If multiple "v=spf1" records are returned, the SPF client MUST reject them all and act as if no version 1 records were returned.

SPF-like records of the form "v=spf1+ext" or "v=spf1.1" are not described by this document.

3.2 SPF Directive Evaluation

There are two types of directives: mechanisms and modifiers. A given mechanism type may always appear multiple times in a record. Modifiers may be constrained to appear at most once per record, depending on the definition of the modifier. Unknown mechanisms cause processing to abort with the result "unknown". Unknown modifiers are ignored by clients.

An SPF record contains an ordered list of mechanisms and modifiers:

```
SPF-record  = version *( 1*SP ( directive / modifier ) ) *SP

version     = "v=spf" 1*DIGIT

prefix      = "+" / "-" / "?" / "~"

directive   = [prefix] mechanism
mechanism   = name [ ":" macro-string ] *[ '/' *DIGIT ]
modifier    = name "=" macro-string
name        = alpha *( alpha / digit / "-" / "_" / "." )
```

Modifiers always contain an equals ('=') character.

Mechanisms usually contain ':' or '/' characters.

Directives that do not contain any of '=', ':', or '/' are mechanisms.

Modifiers MAY appear to the right of a terminal mechanism such as "all". SPF parsers may therefore choose to extract all the modifiers from a record before interpreting mechanisms. Alternatively, they may continue to parse a record in search of a meaningful modifier even after mechanism evaluation has completed.

Each mechanism is considered in turn from left to right.

When a mechanism is evaluated, one of three things can happen: it can match, it can not match, or it can throw an exception.

If it matches, processing ends and the prefix value is returned as the result of that record. (The default prefix value is "+".)

If it does not match, processing continues with the next mechanism. If no mechanisms remain, the default result is specified in section 3.3.

If it throws an exception, mechanism processing ends and the exception value is returned (either "error" indicating a temporary failure, usually DNS-related, or "unknown" indicating a syntax error or other permanent failure resulting in incomplete processing.)

Mechanisms are described in Sections 4 and 5.

A missing prefix for a mechanism is the same as a prefix of "+".

The possible prefixes are:

+	pass
-	fail
~	softfail
?	neutral

Mechanism and modifier names are case-insensitive. A mechanism "INCLUDE" is equivalent to "include". However, case SHOULD be preserved in arguments to mechanisms and modifiers.

3.3 Default result

If none of the mechanisms match and there is no redirect modifier, then the result of the SPF query is "neutral". If there is a redirect modifier, the SPF client proceeds as defined in section 5.1.

Note that SPF records SHOULD always either use a redirect modifier or an "all" mechanism to explicitly terminate processing.

For example:

```
v=spf1 +mx -all

v=spf1 +mx redirect=_spf.example.com
```

4. Mechanism Definitions

This section defines two types of mechanisms.

Basic mechanisms contribute to the SPF language framework. They do not specify a particular type of authentication scheme.

- all
- include

Designated sender mechanisms are used to discover the relationship of the client IP address to the <responsible-sender>.

- a
- mx
- ptr
- ip4
- ip6
- exists

Other mechanisms can be independently defined in the future.

Mechanisms either match, do not match, or throw an exception.
If they match, their prefix value is returned.
If they do not match, processing continues.
If they throw an exception, the exception value is returned.

Several of these mechanisms take an optional <domain-spec> argument. If the <domain-spec> is present, then it is macro expanded (see Section 7) and becomes the <target-name>. If the <domain-spec> is not provided, the <current-domain> is used as the <target-name>.

Several mechanisms require DNS lookups. In those lookups, CNAME responses are followed in the usual way.

4.1 "all"

```
all = "all"
```

The "all" mechanism is a test that always matches. It is used as the rightmost mechanism in an SPF record to provide an explicit default.

For example:

```
v=spf1 +mx +a -all
```

Mechanisms after "all" will never be tested.

4.2 "include"

```
include = "include" ":" domain-spec
```

The "include" mechanism triggers a recursive SPF query. The domain-spec is expanded as per section 7. Then a new query is launched using the resulting string as the <current-domain>. The <responsible-sender> stays the same.

"Include" makes it possible for one domain to designate multiple administratively independent domains.

For example, a vanity domain "example.net" might send mail using the servers of administratively independent domains example.com and example.org.

Example.net could say

"v=spf1 include:example.com include:example.org -all".

That would direct an SPF client to, in effect, search the SPF records for example.com and example.org for a "pass" result. Only if the message were not permitted for either of those domains would the result be "fail".

This mechanism matches when the inner, included query result returns a pass, and doesn't match when the result is fail, softfail, or neutral. However, if the new query returns none, error, or unknown, then processing of the entire SPF query stops immediately and returns the error result.

```
   included    include
   query       mechanism      SPF
   result      result         processing
   -------- -- -------------- ------------------------------------
   pass     => match,         return the prefix value for "include"
   fail     => no match,      continue processing
   softfail => no match,      continue processing
   neutral  => no match,      continue processing
   error    => throw error,   abort processing, return error
   unknown  => throw unknown, abort processing, return unknown
   none     => throw unknown, abort processing, return unknown
```

If the parent domain includes another domain, and that domain one day loses its SPF record, it is better for the query to abort with "unknown" than to continue on to a potential "-all".

The Include mechanism is intended for crossing administrative boundaries. While it is possible to use Includes to consolidate multiple domains that share the same set of designated hosts, domains are encouraged to use Redirects where possible, and to minimize the number of <Includes> within a single administrative domain. For example, if example.com and example.org were managed by the same entity, and if the canonical set of designated mailers for both domains were "mx:example.com", it would be possible for example.org to specify "include:example.com", but it would be preferable to specify "redirect=example.com" or even "mx:example.com".

4.3 Introducing Designated Sender Mechanisms

Designated sender schemes allow SMTP receivers to make policy decisions on the basis of domain name rather than IP address.

These mechanisms allow a domain to declare that certain hosts send mail from that domain. When an SPF client processes these mechanisms, it tests to see if the <sending-host> matches.

Usually, the <sending-host> is the IP address of an SMTP client. The SMTP receiver is the SPF client. The SPF lookup may also operate after the SMTP transaction has terminated. In these cases the <sending-host> may have to be extracted from the Received header or some other meta-data about the message. Received headers can be forged. Still, accurate analysis is possible if care is taken.

If mail is transferred between mail systems internal to an organization, and that organization chooses to process SPF after such transfers, then the <sending-host> should be the external host that first transferred the mail into the organization's mail system.

When the <sending-host> is localhost, Designated Sender mechanisms are not meaningful. Therefore, an SPF client immediately returns "pass" without evaluating mechanisms.

The <sending-host> is required for these mechanisms. If it cannot be determined, then these mechanisms cannot be tested, and "unknown" is returned.

The following conventions apply to designated sender mechanisms:

If the optional <CIDR-length> is given, then only the upper <CIDR-length> bits of each IP are compared to the <sending-host>.

If the SMTP connection is IPv6, read "AAAA lookup" for "A lookup", except where "A" lookups are explicitly specified.

4.4 "a"

This mechanism matches if the <sending-host> is one of the <target-name>'s IP addresses.

A = "a" [":" domain-spec] [dual-cidr-length]

The <sending-host> is compared to the IP address(es) of the <target-name>. If any address matches, the mechanism matches.

4.5 "mx"

This mechanism matches if the <sending-host> is one of the MX hosts for a domain name.

MX = "mx" [":" domain-spec] [dual-cidr-length]

SPF clients first perform an MX lookup on the <target-name>. SPF clients then perform an A lookup on each MX name returned, in order of MX priority. The <sending-host> is compared to each returned IP address. If any address matches, the mechanism matches.

Note Regarding Implicit MXes: If the <target-name> has no MX records, SPF clients MUST NOT pretend the target is its single MX, and MUST NOT default to an A lookup on the <target-name> directly. This behaviour breaks with the legacy "implicit MX" rule. See [RFC2821] Section 5. If such behaviour is desired, the publisher should specify an "a" directive.

4.6 "ptr"

This mechanism tests if the <sending-host>'s name is within a particular domain.

```
     PTR = "ptr" [ ":" domain-spec ]
```

First the <sending-host>'s name is looked up using this procedure: perform a PTR lookup against the <sending-host>'s IP. For each record returned, validate the host name by looking up its IP address. If the <sending-host>'s IP is among the returned IP addresses, then that host name is validated. In pseudocode:

```
  sending-host_names := ptr_lookup(sending-host_IP);
  for each name in (sending-host_names) {
    IP_addresses := a_lookup(name);
    if the sending-host_IP is one of the IP_addresses {
      validated_sending-host_names += name;
  } }
```

Check all validated hostnames to see if they end in the <target-name> domain. If any do, this mechanism matches. If no validated hostname can be found, or if none of the validated hostnames end in the <target-name>, this mechanism fails to match.

Pseudocode:

```
  for each name in (validated_sending-host_names) {
    if name ends in <domain-spec>, return match.
    if name is <domain-spec>, return match.
  }
  return no-match.
```

This mechanism matches if the <target-name> is an ancestor of the <sending-host>, or if the <target-name> and the <sending-host> are the same. For example: "mail.example.com" is within the domain "example.com", but "mail.bad-example.com" is not. If a validated hostname is the <target-name>, a match results.

4.7 "ip4" and "ip6"

These mechanisms test if the <sending-host> falls into a given IP network.

```
 IP4             = "ip4" ":" ip4-network [ ip4-cidr-length ]
 IP6             = "ip6" ":" ip6-network [ ip6-cidr-length ]
 ip4-cidr-length = "/" 1*DIGIT
 ip6-cidr-length = "/" 1*DIGIT

 ip4-network     = dotted-quad notation
 ip6-network     = conventional IPv6 notation
```

The <sending-host> is compared to the given network. If they match, the mechanism matches.

If the cidr-length is omitted, the ip4-cidr-length is taken to be "/32" and the ip6-cidr-length is taken to be "/128".

4.8 "exists"

This mechanism is used to construct an arbitrary host name that is used for a DNS A record query. It allows for complicated schemes involving arbitrary parts of the mail envelope to determine what is legal.

```
     exists = "exists" ":" domain-spec
```

The domain-spec is expanded as per Section 7. The resulting domain name is used for a DNS A lookup. If any A record is returned, this mechanism matches. The lookup type is 'A' even when the connection type is IPv6.

SPF publishers can use this mechanism to specify arbitrarily complex queries. For example, suppose example.com publishes the SPF record:

```
  v=spf1 exists:%{ir}.%{l1r+-}._spf.%{d} -all
```

The target-name might expand to "1.2.0.192.someuser._spf.example.com". This makes fine-grained decisions possible at the level of the user and client IP address.

5. Modifier Definitions

Only two standard modifiers are defined: "redirect" and "exp". SPF clients MUST support them both.

Modifiers are not mechanisms: they do not return match or no-match.

Instead they provide additional information or change the course of SPF processing.

While unrecognized mechanisms cause an immediate "unknown" abort, unrecognized modifiers are simply ignored.

Modifiers therefore provide an easy way to extend the SPF protocol.

This document reserves one extension modifier, "accredit", one deprecated modifier "default", and one future modifier "match_subdomains".

5.1 redirect: Redirected Query

If all mechanisms fail to match, and a redirect modifier is present, then processing proceeds as follows.

```
redirect = "redirect" "=" domain-spec
```

The domain-spec portion of the redirect section is expanded as per the macro rules in section 7. The resulting string is a new domain that is now queried: The <current-domain> is set to this new domain, and the new domain's SPF record is fetched and processed. Note that <responsible-sender> does not change.

The result of this new query is then considered the result of original query.

Note that the newly queried domain may itself specify redirect processing.

This facility is intended for use by organizations that wish to apply the same SPF record to multiple domains. For example:

```
      la.example.com. TXT "v=spf1 redirect=_spf.example.com"
      ny.example.com. TXT "v=spf1 redirect=_spf.example.com"
      sf.example.com. TXT "v=spf1 redirect=_spf.example.com"
    _spf.example.com. TXT "v=spf1 mx:example.com -all"
```

In this example, mail from any of the three domains is described by the same SPF record. This can be an administrative advantage.

Note: in general, a domain A cannot reliably use a redirect to another domain B not under the same administrative control. Since the <responsible-sender> stays the same, there is no guarantee that the SPF directives at domain B will correctly work for addresses in domain A, especially if domain B uses mechanisms involving localparts. An "Include" directive may be more appropriate.

Only one redirect modifier may appear per SPF record. The modifier does not have to appear at the end; it MAY appear anywhere in the record. However, for clarity it is RECOMMENDED that redirect modifiers appear after mechanisms.

5.2 exp: Explanation

The argument to the explanation modifier is a domain-spec to be TXT queried. The result of the TXT query is a macro-string that is macro-expanded. If SPF processing results in a rejection, the expanded result SHOULD be shown to the sender in the SMTP reject message. This string allows the publishing domain to communicate further information via the SMTP receiver to legitimate senders in the form of a short message or URL.

Only one exp modifier may appear per SPF record.

```
explanation = "exp" "=" domain-spec
```

When an SPF client performs a query, and the result is anything other than pass, then the explanation string, if present, SHOULD be presented to the SMTP client after macro expansion. See section 7.

Suppose example.com has this SPF record

```
v=spf1 mx -all exp=explain._spf.%{d}
```

Here are some examples of possible explanation TXT records at explain._spf.example.com:

```
Example.com mail should only be sent by its own servers.
   -- a simple, constant message

%{i} is not one of %{d}'s designated mail servers.
   -- a message with a little more info, including the
   -- SMTP sender's IP address
```

```
See http://%{d}/why.html?s=%{S}&i=%{I}&h=%{H}
   -- a complicated example that constructs a URL with
   -- most of the parameters of the failed message so that
   -- a web page can be generated with instructions
```

If multiple explanation TXT records are returned, they are concatenated in the order they were received. Use of multiple TXT records is discouraged as DNS does not guarantee order.

Note: during recursion into an Include mechanism, explanations do not propagate out. But during execution of a Redirect modifier, the explanation string from the target of the redirect is used.

5.3 accredit: Sender Accreditation

```
accreditation = 'accredit' '=' domain-spec
```

The argument to the accreditation modifier is a domain-spec to be macro-expanded and queried. The result of the query is interpreted according to the definitions set forth by the accreditation service.

For example,

```
accredit=%{d}.accreditation-provider.example.com
accredit=%{ir}.accreditation-provider.example.com
```

This facility allows the publishing domain to make independently verifiable assertions about itself in machine-readable form.

Multiple "accredit" modifiers may appear in one SPF record.

The "accredit" modifier is OPTIONAL. SPF publishers MAY omit it. SPF clients MAY ignore any or all "accredit" modifiers. If a receiver does not recognize the domain-spec argument, it MAY ignore the modifier.

It is expected that SPF-enabled receivers will maintain a library of recognized accreditation providers, keyed by the domain-spec. An accreditation provider is responsible for describing the protocol it uses to encode assertions. For example, suppose an accreditation provider supports DNS "A" queries against the expanded domain-spec. A result of NXDOMAIN could mean "domain is not known to the accreditation service." A result of "127.0.0.10" could mean "the accreditation service vouches for the integrity of the sender domain." Accreditation providers can make up any protocol they like as long as they can convince receivers to use it.

Accreditation is only meaningful if the result of the SPF query is a PASS.

Accreditation operates on behalf of the sender. Receivers, and the reputation services that operate on their behalf, are expected to adopt a critical stance toward accreditation assertions.

6. Miscellaneous

6.1 Unrecognized Mechanisms and Modifiers

Future extensions to this standard may introduce new mechanisms and modifiers.

Unrecognized mechanisms cause processing to abort: if, during evaluation of an SPF record, an SPF client encounters a mechanism which it does not understand or which it cannot properly evaluate (due perhaps to insufficient information about the message at evaluation time), then it terminates processing and returns "unknown", without evaluating any further mechanisms. Mechanisms listed before the unknown mechanism MUST, however, be evaluated.

For example, given the record

```
v=spf1 a mx ptr domainkeys:_dk.%{d} -all
```

messages that match the "a", "mx", or "ptr" mechanisms would return a "pass" result. An SPF client that did not recognize the mechanism "domainkeys" would return "unknown". An SPF client that was domainkeys-aware would be able to perform extended evaluation. If the message matched the domainkeys test, it would pass; if it did not, evaluation would proceed to "-all" and return "fail".

"domainkeys" is an example of an unknown extension mechanism that could be defined in future versions of this standard. It is NOT defined by this proposal.

Unrecognized modifiers are ignored: if an SPF client encounters modifiers which it does not recognize, it MUST ignore them and continue processing. Modifiers always contain an "=" sign.

Unrecognized mechanisms are preserved in the Received-SPF header. See section 6.3.

6.2 Processing Limits

During processing, an SPF client may perform additional SPF subqueries due to the Include mechanism and the Redirect modifier.

SPF clients must be prepared to handle records that are set up incorrectly or maliciously. SPF clients MUST perform loop detection, limit SPF recursion, or both. If an SPF client chooses to limit recursion depth, then at least a total of 20 redirects and includes SHOULD be supported. (This number should be enough for even the most complicated configurations.)

If a loop is detected, or if more than 20 subqueries are triggered, an SPF client MAY abort the lookup and return the result "unknown".

Regular non-recursive lookups due to mechanisms like "a" and "mx" or due to modifiers like "exp" do not count toward this total.

6.3 The Received-SPF header

It is RECOMMENDED that SMTP receivers record the result of SPF processing in the message headers. If an SMTP receiver chooses to do so, it MUST use the "Received-SPF" header defined here. This information is intended for the recipient. (Information intended for the sender of the e-mail is described in Section 5.2, Explanation.)

The header SHOULD be prepended to existing headers. It MUST appear above any other Received-SPF headers in the message. The header has the format:

```
header = "Received-SPF:" 1*SP result [ 1*SP "(" comment ")" ]
         *( 1*SP key-value-pair )

result = "pass" / "fail" / "error" / "softfail" / "neutral" /
         "none" / "unknown" / unknown-mechanisms

unknown-mechanisms = "unknown" *( 1*SP [prefix] mechanism )

key-value-pair = 1*VCHAR "=" *(WSP / VCHAR) ";"

comment = [ smtp-receiver-hostname ": " comment-string ]
```

The comment-string should convey supporting information for the result (such as <responsible-sender> and <current-domain>).

If processing was aborted due to unrecognized mechanisms, the Received-SPF header SHOULD show the unrecognized mechanisms after the "unknown" word.

Example headers generated by mybox.example.org:

```
Received-SPF: pass (mybox.example.org: domain of
   myname@example.com designates 192.0.2.1 as permitted sender)
      receiver=mybox.example.org; client-ip=192.0.2.1;
      envelope-from=<myname@example.com>; helo=foo.example.com;

Received-SPF: fail (mybox.example.org: domain of
                    myname@example.com does not designate
                    192.0.2.1 as permitted sender)
                    receiver=mybox.example.org;
                    client-ip=192.0.2.1;
                    envelope-from=<myname@example.com>;
                    helo=foo.example.com;

Received-SPF: softfail (mybox.example.org: domain of
                        transitioning myname@example.com does not
                        designate 192.0.2.1 as permitted sender)

Received-SPF: neutral (mybox.example.org: 192.0.2.1 is neither
                       permitted nor denied by domain of
                       myname@example.com)

Received-SPF: none (mybox.example.org: myname@example.com does
                    not designate permitted sender hosts)

Received-SPF: unknown -extension:foo (mybox.example.org: domain
                       of myname@example.com uses a
                        mechanism not recognized by this client)

Received-SPF: error (mybox.example.org: error in processing
                     during lookup of myname@example.com: DNS
                     timeout)
```

SPF clients may append zero or more of the following key-value-pairs at their discretion:

receiver	the hostname of the SPF client
client-ip	the IP address of the SMTP client
envelope-from	the envelope sender address
helo	the hostname given in the HELO or EHLO command
mechanism	the mechanism that matched (if no mechanisms matched, substitute the word "default".)
problem	if an error was returned, details about the error

Other key-value pairs may be defined by SPF clients. Until a new key name becomes widely accepted, new key names should start with "x-".

7. Macros

7.1 Macro definitions

Certain directives perform macro interpolation on their arguments.

```
  macro-string = *( macro-char / VCHAR )
  macro-char   = ( "%{" ALPHA transformer *delimiter "}" )
                 / "%%" / "%_" / "%-"
  transformer  = [ *DIGIT ] [ "r" ]
  delimiter    = "." / "-" / "+" / "," / "/" / "_" / "="
```

A literal "%" is expressed by "%%".
%_ expands to a single " " space.
%- expands to a URL-encoded space, viz. "%20".

The following macro letters are expanded in directive arguments:

```
   l = local-part of responsible-sender
   s = responsible-sender
   o = responsible-domain
   d = current-domain
   i = SMTP client IP (nibble format when an IPv6 address)
   p = SMTP client domain name
   v = client IP version string: "in-addr" for ipv4 or "ip6" for ipv6
   h = HELO/EHLO domain
   r = receiving domain
```

The following macro letters are expanded only in "exp" text:

```
   c = SMTP client IP (easily readable format)
   t = current timestamp in UTC epoch seconds notation
```

The uppercase versions of all these macros are URL-encoded.

A '%' character not followed by a '{', '%', '-', or '_' character
MUST be interpreted as a literal. SPF publishers SHOULD NOT rely on
this feature; they MUST escape % literals. For example, an
explanation TXT record

```
   Your spam volume has increased by 581%
```

is incorrect. Instead, say

```
   Your spam volume has increased by 581%%
```

Legal optional transformers are:

```
     *DIGIT ; zero or more digits
     'r'    ; reverse value, splitting on dots by default
```

If transformers or delimiters are provided, the macro strings are split into parts. After performing any reversal operation or removal of left-hand parts, the parts are rejoined using "." and not the original splitting characters.

By default, strings are split on "." (dots). Macros may specify delimiter characters which are used instead of ".". Delimiters MUST be one or more of the characters:

"." / "-" / "+" / "," / "/" / "_" / "="

The 'r' transformer indicates a reversal operation: if the client IP address were 192.0.2.1, the macro %{i} would expand to "192.0.2.1" and the macro %{ir} would expand to "1.2.0.192".

The DIGIT transformer indicates the number of right-hand parts to use after optional reversal. If a DIGIT is specified, it MUST be nonzero. If no DIGITs are specified, or if the value specifies more parts than are available, all the available parts are used. If the DIGIT was 5, and only 3 parts were available, the macro interpreter would pretend the DIGIT was 3. Implementations MAY limit the number, but MUST support at least a value of 9.

For the "l" and "s" macros: when the local-part is not defined, the string "postmaster" is substituted. The local-part might be undefined if the <current-domain> is drawn from the HELO command rather than the MAIL FROM.

For IPv4 addresses, both the "i" and "c" macros expand to the standard dotted-quad format.

For IPv6 addresses, the "i" macro expands to dot-format address; it is intended for use in %{ir}. The "c" macro may expand to any of the hexadecimal colon-format addresses specified in [RFC3513] section 2.2. It is intended for humans to read.

Use of the "t" macro in DNS lookups would greatly reduce the effectiveness of DNS caching. The "t" macro is only allowed in explanation records. The value of the "t" macro SHOULD NOT change during the evaluation of a given SPF record.

The "p" macro expands to the validated domain name of the SMTP client. The validation procedure is described in section 4.6. If there are no validated domain names, the word "unknown" is substituted. If multiple validated domain names exist, the first one returned in the PTR result is chosen.

The "r" macro expands to the name of the receiving MTA. This SHOULD be a fully qualified domain name, but if one does not exist (as when the checking is done by a script) or if policy restrictions dictate otherwise, the word "unknown" SHOULD be substituted. The domain name MAY be different than the name found in the MX record that the client MTA used to locate the receiving MTA.

The "s" macro expands to the sender email address: a localpart, an @ sign, and a domain. The "o" macro is the domain part of the "s". They remain the same during a recursive "include" or "redirect" subquery.

When the result of macro expansion is used in a domain name query, if the expanded domain name exceeds 255 characters (the maximum length of a domain name), the left side is truncated to fit, by removing successive subdomains until the total length falls below 255 characters.

Uppercased macros are URL escaped.

URL encoding is described in [RFC2396].

7.2 Expansion Examples

The <responsible-sender> is strong-bad@email.example.com.
The IPv4 SMTP client IP is 192.0.2.3.
The IPv6 SMTP client IP is 5f05:2000:80ad:5800::1.
The PTR domain name of the client IP is mx.example.org.

macro	expansion
%{s}	strong-bad@email.example.com
%{o}	email.example.com
%{d}	email.example.com
%{d4}	email.example.com
%{d3}	email.example.com
%{d2}	example.com
%{d1}	com
%{p}	mx.example.org
%{p2}	example.org
%{dr}	com.example.email
%{d2r}	example.email
%{l}	strong-bad
%{l-}	strong.bad
%{lr}	strong-bad
%{lr-}	bad.strong
%{l1r-}	strong

```
macro-string                                    expansion
-------------------------------------------------------------------------
%{ir}.%{v}._spf.%{d2}              3.2.0.192.in-addr._spf.example.com
%{lr-}.lp._spf.%{d2}                   bad.strong.lp._spf.example.com

%{lr-}.lp.%{ir}.%{v}._spf.%{d2}
                     bad.strong.lp.3.2.0.192.in-addr._spf.example.com

%{ir}.%{v}.%{l1r-}.lp._spf.%{d2}
                          3.2.0.192.in-addr.strong.lp._spf.example.com

%{p2}.trusted-domains.example.net
                                example.org.trusted-domains.example.net

IPv6:
%{ir}.example.org              1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.8.
                                5.d.a.0.8.0.0.0.2.5.0.f.5.example.org
```

8. Conformance Definitions

The following sections define SPF conformance applied to various entities.

8.1 Introduction

In an SMTP+SPF transaction, there are three primary entites:

- the envelope sender domain
- the sending host (the SMTP client)
- the SMTP receiver

The conformance status of each entity has implications for the transaction as a whole.

8.2 Conformance with regard to sender domains

For a domain to be considered SPF-conformant, its authoritative DNS servers are REQUIRED to publish an SPF record for that domain. Domains which do not publish SPF data SHALL NOT be deemed SPF-conformant.

If foo.example.com claims SPF compliance, it must have a record of the form:

```
foo.example.com IN TXT "v=spf1 ..."
```

A domain also SHOULD publish policy records for each of its designated servers.

8.3 Conformance with regard to sending e-mail systems

To be considered SPF-conformant, an SMTP sending host MUST resolve a "pass" for all the SPF-conformant domains which appear in the "MAIL FROM" command.

An SMTP sending host MUST also resolve a "pass" for all the SPF-conformant domains which appear in the "HELO" or "EHLO" command.

If the domains used in MAIL FROM and in HELO/EHLO do not publish SPF information, an SMTP sending host is considered conformant by default. Only when those domains do publish SPF is the SMTP sending host required to resolve a "pass".

For example: in a transaction where host mx01 emits:

```
HELO mx01.example.com
MAIL FROM:<strongbad@example.com>
```

An SMTP+SPF receiver will attempt to find an SPF record at

```
example.com TXT
```

An SMTP+SPF receiver may also attempt to find an SPF record at

```
mx01.example.com TXT
```

If either query returns an SPF record, host mx01 MUST return a "pass" for that SPF test.

8.4 Conformance with regard to receiving e-mail systems

To describe itself as SPF-conformant, an SMTP receiver MUST perform SPF tests.

SPF tests need not be performed while an SMTP transaction is ongoing: if the MDA performs the test, that is sufficient. A server need not reject a message; but if it does not, it SHOULD add a Received-SPF header. If a server rejects a message, it SHOULD include any <explanation> provided by the SPF publisher.

If a receiver system has a choice of testing the envelope sender (as recorded in the Return-Path header) versus the message headers (as recorded in Sender or From), the envelope is recommended.

If the domain passed in the HELO command is a fully-qualified domain name, an SPF-conformant receiver MAY test that domain name.

Receiver systems SHOULD exclude special recipients such as postmaster@ and abuse@ from SPF processing. See [RFC2142].

SPF is only one component in a policy engine. An SPF-conformant SMTP receiver is NOT REQUIRED to perform SPF tests on messages whose dispositions have already been decided on the basis of other policy.

> Example 1: if an SMTP receiver requires that sender domains must possess MX or A records, and rejects transactions where they do not, then SPF tests are moot.
>
> Example 2: if an SMTP receiver expects messages from a trusted client, such as a secondary MX for its own domain, then SPF tests are not needed.
>
> Example 3: if an SMTP receiver is considering a transaction which does not yield a fully-qualified domain name in either the MAIL FROM sender or the HELO command, SPF tests are not appropriate, and the disposition of the message should be decided on the basis of other policy.

8.5 Conformance with regard to a particular SMTP transaction

An email message during delivery is conformant if SPF evaluation results in a "pass", "neutral", or "softfail".

8.6 Conformance with regard to an email-sending user

An email-sending user is conformant if all of her outbound mail is sent through a designated mailer for her sender domain.

8.7 Rejection of non-SPF conformant email

Mail from a domain SHOULD NOT be automatically treated as suspect just because the domain doesn't publish SPF records.

If SPF tests return an explicit "fail" result during processing, the receiver domain MAY reject, label, or classify the message as it wishes. If the message is rejected, the receiver domain SHOULD provide the "exp" string specified in section 5.2.

8.8 Rejection of SPF conformant email

An SPF email system MAY choose to reject or discard email on the basis of local policy. SPF is one component in an overall email-policy engine. SPF merely makes it possible for policy decisions to be made with confidence at the sender-domain level. The actual policy decisions are outside the scope of this document.

8.9 Recommendations

If a domain contains subdomains or hostnames that have A or MX records, those subdomains and hostnames SHOULD publish SPF records as well. If they do not, they remain at risk of forgery.

Domain names used in a legitimate envelope sender SHOULD possess MX records.

This proposal recommends the deprecation of the legacy "implicit MX" rule defined in [RFC2821] section 5. Domains that want to receive mail SHOULD always define explicit MX records.

This proposal also recommends that HELO arguments SHOULD be fully-qualified domain names that resolve to the IP address of the sending MTA.

8.10 Changes to Existing Semantics

8.10.1 The Return-Path is now also a Responsible Sender

From [RFC2821]:

> The <reverse-path> portion of the first or only argument contains the source mailbox (between "<" and ">" brackets), which can be used to report errors (see section 4.2 for a discussion of error reporting).

When SPF is used to authenticate the return-path, the domain in the source mailbox is now also considered accountable for injecting the message into the mailstream.

This semantic change is justified by the desire to control joe-jobs. Joe jobs are a distributed denial of service attack against a given address executed by forging messages using a victim sender address and sending them to thousands of recipients. Inevitably, some of those delivery attempts fail, and bounce messages are generated to the victim sender address. These unwanted bounce messages can end up crippling the victim mailbox. SPF gives these potential victims a way to protect their mailboxes. With SPF, senders can now control the use of their address in the return-path.

9. Applicability Statement

This section discusses deployment considerations.

9.1 Adoption by disreputable domains

It is trivial for a domain to publish "+all" to allow all. A disreputable domain could then send unwanted email from any host. This is a common objection to SPF.

SPF provides a value-neutral framework for sender authentication and accreditation. Senders use that framework to make assertions regarding the quality of messages.

It is the responsibility of the SMTP receiver to evaluate those assertions. First the SMTP receiver discards obvious forgeries. Then it evaluates the remaining messages according to the reputation of the sender and any accompanying accreditation assertions.

Whether a given domain is reputable or not is a decision that belongs to an SMTP receiver. Policy decisions regarding particular messages are outside the scope of SPF.

9.2 Limitations

In an SPF-conformant environment, envelope sender forgery is limited to the local domain. Unless per-user macros are used, it is possible for one user to forge another user's address within that domain. However, the organization responsible for the domain presumably has the wherewithal to follow an audit trail.

9.3 Phased Rollout

At an adopting domain, adoption of SPF could occur in phases. A domain might move through these phases by changing its default response type from "neutral" to "softfail" to "fail".

The phases are characterized by different levels of awareness among the domain's userbase, and different levels of strictness on the part of SPF-conformant receivers.

When a sufficient majority of its users are SPF-conformant, a domain SHOULD change its default to "-all". This constitutes a request to mail receivers to reject non-conformant mail. Setting "-all" protects the users from account fraud and joe-jobbing.

Messages that explicitly fail SPF with a "fail" SHOULD be rejected.

9.4 Verbatim Forwarding

SPF changes the model of email from a store-and-forward system to an end-to-end system. Verbatim email forwarding suffers accordingly.

In verbatim forwarding, an intermediate forwarding host rewrites the envelope recipient, but leaves the envelope sender untouched.

There are two ways to preserve forwarding functionality under SPF. Either forwarders can change to remailing, or receiver systems can whitelist forwarders.

In remailing, forwarding hosts rewrite the return-path address. The rewritten envelope sender can take a variety of forms, depending on whether bounces should be forwarded back to the original sender. If bounces are unimportant, the rewritten sender could be as simple as <nobody@example.com>. If bounces are important, forwarders can use envelope encapsulation with a MAC nonce, or rewrite the address into a database-backed cookie.

If return-path rewriting is not feasible, receiver systems can simply whitelist trusted sites which are known to forward mail to local receivers.

9.5 Per-user exemptions

The "exists" mechanism can be used to exempt certain users from the SPF requirements that apply to the rest of the domain.

10. Security Considerations

SPF depends on DNS. A malicious attacker could poison a target's DNS cache with spoofed DNS data.

SPF assumes the client IP address is true. A malicious attacker could spoof TCP sequences to make mail appear to come from a designated host. If this happens, an "echo" command could be introduced to ESMTP to require a simple challenge/response confirmation.

SPF clients need to limit the number of includes and redirects to avoid attacks. Implementations are required to follow recursion to a minimum of 20.

11. IANA Considerations

The registry of standard mechanism and modifier names may be turned over to IANA for management.

12. Contributors and Acknowledgements

SPF owes a debt of parentage to RMX (by Hadmut Danisch in 2003) and to DMP (by Gordon Fecyk in 2003). It traces its ancestry farther back through "Repudiating Mail-From" by Paul Vixie in 2002 to a suggestion by Jim Miller in 1998.

Philip Gladstone contributed macros to the specification, multiplying the expressiveness of the language and making per-user and per-IP lookups possible.

The authors would also like to thank the following individuals who contributed to, critically reviewed, or otherwise furthered the development of this specification. The authors wish to apologize in advance for any omissions.

Jameel Akari, Marc Alaia, Dave Alden, Tom Allison, Eric Allman, Justo Alonso, Mark Anderson, Matthias Andree, Don Andrews, Mohammad Arca, Aredridel, William Astle, Bob Atkinson, Roy Badami, Andy Bakun, Derek J. Balling, Arik Baratz, Graham Barr, Matthew Barr, Ernesto Baschny, Mike Batchelor, Peter Baumann, Steven Bellovin, Jon Bertrand, Paul Blair, Andrew Boling, Richard Bollinger, Dan Boresjo, Nicolas Bougues, Daniel Bourque, Jeremy T. Bouse, Raymond S Brand, Seth Breidbart, David Brodbeck, Neil Brown, Zack Brown, Michael R. Brumm, Jason Buchanan, Jasmin Buchert, Ted Cabeen, Dave Camp, Anthony Campbell, Bryan Campbell, Brian Candler, John Capo, Dennis Carr, L. Carver, Lee Carver, Ian Castle, Jose Celestino, Christopher Chan, Jason Chen, Joe Christy, Andrew Church, Greg Cirino, Bradley Cloete, James H. Cloos Jr., Bill Cole, Brian Coloney, Sean Comeau, Greg Connor, Erik Corry, Dj Coster, James Couzens, Chris Cowherd, Dave Crocker, Arlie Davis, Alan DeKok, Mark Delany, Kitt Diebold, Mark Jason Dominus, Andrew W. Donoho, William H. Dorell, Jeremy Doupe, Lilia Downs, Chris Drake, Jesus Duarte, Viktor Duchovni, Lars B. Dybdahl, David Dyer-Bennet, Lyndon Eaton, Henrik Edlund, William Elan, Matthew Elvey, Stefan Engelbert, Shaun T. Erickson, Ray Everett-Church, Mark Farver, Nick Fields, Guillaume Filion, Tony Finch, Cary Fitch, Gustav Foseid, Mark Foster, Steven Foster, Benjamin Franz, Tim Freeman, Tugrul Galatali, Stuart D. Gathman, Fotis Georgatos, Richard George, Oliver Gerler, Dean Gibson, B. Gingery, Eric Girard, Tim Gladding, Philip Gladstone, Etta Good, Seth Goodman, Gabriel Granger, Ben Greear, Bob Greene, Ronald F. Guilmette, Phillip Hallam-Baker, Clifford Hammerschmidt, Catherine Hampton,

Ask Bjoern Hansen, Richard Hansen, Howard Lee Harkness, Thomas Harold, Edward Ned Harvey, Ned Harvey, Brian Hatch, Refugio Hayden, Philip Hazel, Zan Hecht, Eric Helfgott, George Herson, Greg Hewgill, Alan Hodgson, Vidar Holen, Sam Horrocks, Phil Howard, Paul Howarth, Anthony Howe, Marc Hudson, John Hughes, Kenn Humborg, Adam Hunt, Carl Hutzler, Paul Iadonisi, Lars Magne Ingebrigtsen, Aditya Ivaturi, Mitcheal Jackmoore, Jeremy Jackson, Bryce Jasmer, Mark Jeftovic, Jim Jobe, B. Johannessen, Thomas H Jones II, Nico Kadel-Garcia, Rob Kaper, Phil Karn, Harry Katz, Lou Katz, Richard Kay, Izzy Kindred, Alain Knaff, Don Koch, Kevin Kolk, Tomasz Konefal, Thor Kooda, Karl Kraft, Andreas Kreuzinger, Russell Kroll, Carsten Kuckuk, Jon Kyme, Ty Lammy, Bill Landry, Mark Lentczner, Nate Leon, Andy Lester, John R Levine, Jon Loeliger, Will Lowe, Dave Lugo, Jim Lyon, Marty Lyons, Vivien M., Daniel Mack, Dr Belo Madu, Lee Maguire, Ryan Malayter, Walt Mankowski, Marrandy, K.F.J. Martens, John A. Martin, Tracy Martin, Justin Mason, Jeroen Massar, Mike McCandless, Joel McClung, Chuck Mead, Tim Meadowcroft, Julian Mehnle, Cyrus Mehta, Michael Meier, Scott Merrill, Nigel Metheringham, Bob Miller, Nelson Minica, Anne Mitchell, George Mitchell, Dr. Ernst Molitor, Michael Fischer V. Mollard, Philipp Morger, Roger Moser, der Mouse, Matthew Mucker, Simon J Mudd, Graham Murray, Dan Nadir, Alain Nakache, Brian Nelson, Scott Nelson, Sam Norris, Daryl Odnert, Leonard Orb, Steven W. Orr, Seun Osewa, Gerald Oskoboiny, Harry Palmer, John Payne, Hans Dieter Pearcey, Randy Pearson, Matt Perry, R. Scott Perry, Kevin Peuhkurinen, Nick Phillips, Richard Pitt, Jordan Pollack, Bob Poortinga, Jim Popovitch, Kenneth Porter, Bob Proulx, Loic Prylli, Rich Puhek, Nils Puhlmann, James Pullicino, Jeremy Pullicino, Daniel Quinlan, Ramakanta, Suresh Ramasubramanian, Jim Ramsay, Eric S. Raymond, Kevin Reed, Alan Reider, Joe Rhett, Bill Rockefeller, Daniel Roethlisberger, Andrew Rose, Alex Rosen, David Saez, Hector Santos, Christophe Saout, Scott Savarese, Wayne Schlitt, George Schlossnagle, Theo Schlossnagle, Stuart Schneider, Neil Schwartzman, Frank Segtrop, Klaus Alexander Seistrup, Will Senn, Matt Sergeant, Yakov Shafranovich, Shevek, Mark Shewmaker, G. Roderick Singleton, Fridrik Skulason, Martin H. Sluka, Andy Smith, Karl J. Smith, Larry Smith, Steven Earl Smith, Richard Soderberg, Rolf E. Sonneveld, Robert Spier, Jonathan Steinert, Thomas R. Stephenson, Rick Stewart, Andrew Sweger, Bob Tanner, Daniel Taylor, Brad Templeton, Rahul Tongia, Laszlo Toth, Dustin D. Trammell, Mark Tranchant, Philip Tucker, Alex Van Den Bogaerdt, Dirk Van Mieghem, Rik Van Riel, Theo Van Dinter, Wietse Venema, Kelson Vibber, Peter Viertel, Paul Vixie, Martin Treusch Von Buttlar, Graham Wager, Matthew Walker, Jasper Wallace, John Warren, Odhiambo Washington, Terence Way, George Webb, Wechsler, Rick Wesson, Casey West, Peter Westlake, Nathan Wharton, David A. Wheeler, Weldon Whipple, Phil White, Sanford Whiteman, Colin Whittaker, Tim Wilde, Jan Wildeboer, Dan Willett Sr.,

Steven G. Willis, Chuck Wolber, David Woodhouse, Simon Woodward, Greg Wooledge, Paul Wouters, Samson Yormie, Zolta'N Za'Mbori, Joe Zasada, and Lloyd Zusman.

The folks on the SPAM-L mailing list.
The folks on the SPAM-L mailing list.
The folks on the ASRG and MARID/MXCOMP mailing lists.
The folks on the spf-discuss mailing list.
The folks on the mailing list that shall not be named.
The folks on #perl.

Appendix A. Collected ABNF for SPF records

This section is normative and any discrepancies with the ABNF fragments in the preceding text are to be resolved in favor of this grammar.

See [RFC2234] for ABNF notation.

```
SPF-record  = version *( 1*SP ( directive / modifier ) ) *SP

version     = "v=spf" 1*DIGIT

directive   = [ prefix ] mechanism
prefix      = "+" / "-" / "?" / "~"
modifier    = redirect / explanation / unknown-modifier
redirect    = "redirect" "=" domain-spec
explanation = "exp" "=" domain-spec
unknown-modifier = name "=" macro-string

mechanism   = ( all / include
              / A / MX / PTR / IP4 / IP6 / exists
              / extension )

all          = "all"
include      = "include"  ":" domain-spec
A            = "a"      [ ":" domain-spec ] [ dual-cidr-length ]
MX           = "mx"     [ ":" domain-spec ] [ dual-cidr-length ]
PTR          = "ptr"    [ ":" domain-spec ]
IP4          = "ip4"      ":" ip4-network   [ ip4-cidr-length ]
IP6          = "ip6"      ":" ip6-network   [ ip6-cidr-length ]
exists       = "exists"   ":" domain-spec

extension    = name [ ":" macro-string ]

ip4-network  = as in [RFC2373] [15], e.g. 192.0.2.0
ip6-network  = as in [RFC2373] [15], e.g. 12AB:0:0:CD30

domain-spec  = domain-name / macro-string
domain-name  = domain-part *( "." domain-part ) [ "." ]
domain-part  = as defined in [RFC1034]

dual-cidr-length = [ ip4-cidr-length ] [ "/" ip6-cidr-length ]
ip4-cidr-length  = "/" 1*DIGIT
ip6-cidr-length  = "/" 1*DIGIT

macro-string = *( macro-char / VCHAR )
macro-char   = ( "%{" ALPHA transformer *delimiter "}" )
               / "%%" / "%_" / "%-"
transformer  = [ *DIGIT ] [ "r" ]
```

```
    name        = alpha *( alpha / digit / "-" / "_" / "." )
    delimiter   = "." / "-" / "+" / "," / "/" / "_" / "="
```

Appendix B. Extended Examples

These examples are based on the following DNS setup:

```
    ; A domain with two mail servers, two hosts
    ; and two servers at the domain name

    $ORIGIN example.com.
    @           MX  10 mail-a
                MX  20 mail-b
                A   192.168.1.10
                A   192.168.1.11
    amy         A   192.168.1.65
    bob         A   192.168.1.66
    mail-a      A   192.168.1.129
    mail-b      A   192.168.1.130
    www         CNAME example.com.

    ; The reverse IP for that domain

    $ORIGIN 1.168.192.in-addr.arpa.
    10          PTR example.com.
    11          PTR example.com.
    65          PTR amy.example.com.
    66          PTR bob.example.com.
    129         PTR mail-a.example.com.
    130         PTR mail-b.example.com.

    ; A related domain

    $ORIGIN example.org
    @           MX  10 mail-c
    mail-c      A   192.168.2.140

    ; The reverse IP for that domain

    $ORIGIN 2.168.192.in-addr.arpa.
    140        PTR mail-c.example.org.

    ; A rogue reverse IP domain that claims to be
    ; something it's not

    $ORIGIN 0.0.10.in-addr.arpa.
    4           PTR bob.example.com.
```

Appendix B.1 Simple Example

```
   If <current-domain> is "example.com", then this describes the
   effect various possible SPF records for example.com would have
   on various <sending-hosts>

   "v=spf1 +all"
        -- any mail message passes


   "v=spf1 a -all"
        -- sending hosts 192.168.1.10 and 192.168.1.11 pass

   "v=spf1 a:example.org -all"
        -- no sending hosts pass since example.org has no A records


   "v=spf1 mx -all"
        -- sending hosts 192.168.1.129 and 192.168.1.130 pass

   "v=spf1 mx:example.org -all"
        -- sending host 192.168.2.140 passes

   "v=spf1 mx mx:example.org -all"
        -- sending hosts 192.168.1.129, 192.168.1.130,
           and 192.168.2.140 pass

   "v=spf1 mx/24 mx:example.org/24 -all"
        -- any sending host in 192.168.1.0/24 or 192.168.2.0/24 passes


   "v=spf1 ptr -all"
        -- sending host 192.168.1.65 passes
                (reverse IP is valid and in example.com)
        -- sending host 192.168.2.140 fails
                (reverse IP is valid, but not in example.com)
        -- sending host 10.0.0.4 fails
                (reverse IP is not valid)


    "v=spf1 ip4:192.168.1.128/25 -all"
        -- sending host 192.168.1.65 fails
        -- sending host 192.168.1.129 passes
```

Appendix B.2 Multiple Domain Examples

These examples show the effect of related SPF records:

```
example.org: "v=spf1 include:example.com include:example.net -all"
```

This SPF record would be used if mail from example.org actually came through servers at example.com and example.net. Example.org's designated servers are the union of example.com and example.net's designated servers.

```
1.example.org: "v=spf1 redirect=example.org"
2.example.org: "v=spf1 redirect=example.org"
3.example.org: "v=spf1 redirect=example.org"
```

These SPF records allow a set of domains that all use the same mail system to make use of that mail system's SPF record. In this way, only the mail system's SPF record needs to updated when the mail setup changes. These domains' SPF records never have to change.

Appendix B.3 RBL Style Example

Imagine that, in addition to the domain records listed above, there are these:

```
$Origin _spf.example.com.
mary.mobile-users                 A 127.0.0.2
fred.mobile-users                 A 127.0.0.2
15.15.168.192.joel.remote-users   A 127.0.0.2
16.15.168.192.joel.remote-users   A 127.0.0.2
```

The following SPF records describe users at example.com who mail from arbitrary servers, or who mail from personal servers.

```
example.com:
    "v=spf1 mx
            include:mobile-users._spf.%{d}
            include:remote-users._spf.%{d} -all"

mobile-users._spf.example.com:
    "v=spf1 exists:%{l1r+}.%{d}"

remote-users._spf.example.com:
    "v=spf1 exists:%{ir}.%{l1r+}.%{d}"
```

Normative References

[RFC2396]

Informative References

[RFC1034]
[RFC1464]
[RFC2119]
[RFC2142]
[RFC2234]
[RFC2373]
[RFC2505]
[RFC2821]
[RFC2822]

Danisch, Hadmut.
"The RMX DNS RR Type for light weight sender authentication",
http://www.danisch.de/work/security/antispam.html, October 2003,
Work In Progress.

Fecyk, Gordon. "Designated Mailers Protocol", December 2003,
Work In Progress. http://www.pan-am.ca/dmp/

Wong, M.W., "Sender Rewriting Scheme", Work In Progress,
http://spf.pobox.com/srs.html

DeKok, Alan. The LMAP discussion document.

Vixie, Paul. "Repudiating Mail-From".
http://ops.ietf.org/lists/namedroppers/namedroppers.2002/msg00658.html

http://nospam.couchpotato.net/

http://asrg.sp.am/

http://www.ietf.org/html.charters/marid-charter.html

Authors

Meng Weng Wong
Singapore
mengwong+spf@pobox.com

Mark Lentczner
1209 Villa Street
Mountain View, CA 94041
United States of America
markl@glyphic.com

Comments on this draft are welcome. In the interests of openness, before contacting the authors directly, please post to the spf-discuss mailing list.

To join the mailing list, please see
http://spf.pobox.com/mailinglist.html

Developers of SPF-conformant software SHOULD join the spf-devel mailing list. They also SHOULD consult the SPF Developers Guide at http://spf.pobox.com/developers-guide.html. While specifications published as RFCs are relatively static, the mailing list and developers guide are living resources. They augment this core protocol specification with generally accepted implementation practices which are outside the scope of this document. They also provide a forum for interoperability testing.