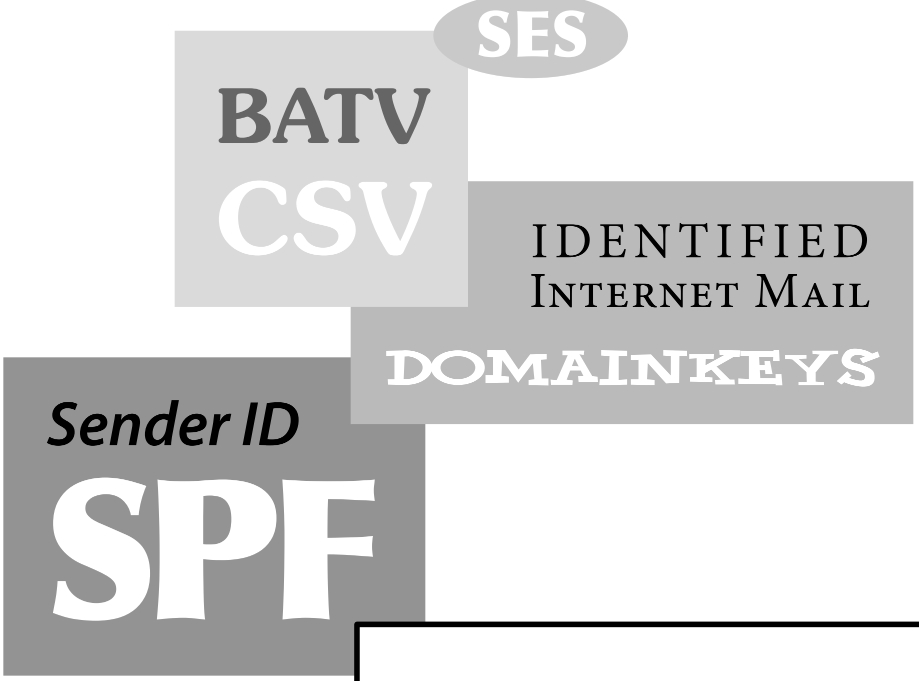


Sender Authentication

What To Do

by Meng Weng Wong

A Messaging Anti-Abuse Working Group White Paper

Contents

A MAAWG White Paper on
Sender Authentication Deployment

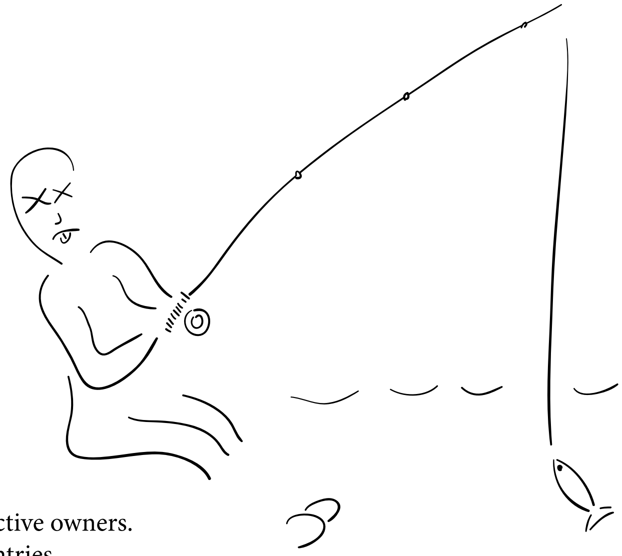
http://spf.pobox.com/whitepaper.pdf

Meng Weng Wong
mengwong@pobox.com

Senior Technical Advisor
Messaging Anti-Abuse Working Group

Founder & CTO for Special Projects
pobox.com

December 2 2004 · release 1



Zombie Phishing

Opinions expressed in this white paper are those of the author and
do not necessarily reflect the opinions of MAAWG or MAAWG's member organizations.

In proportion as our inward life fails, we go more constantly and desperately to the post-office. You may depend on it, that the poor fellow who walks away with the greatest number of letters, proud of his extensive correspondence, has not heard from himself this long while.

Life Without Principle
Henry David Thoreau

Introduction

2004 saw a great deal of hue and cry about spam. Now the dust is beginning to settle. A number of complementary technologies are still standing. They will be rolled out in 2005. This white paper explains why they're important, how they work, and how you can put them to use.

The email infrastructure as originally designed lacks a critical element: sender authentication. Sender authentication gives computers the ability to tell whether a message is forged or authentic. Without that ability, receiver systems cannot easily detect and block forgeries at SMTP time. Adding that ability sets the stage for complementary technologies — reputation and accreditation systems. Together, these technologies make it possible to build a spam-free layer on top of the existing email system.

Sender authentication protocols will be widely deployed in 2005. This document explains why they are important, how they work, and how you can deploy them. It pays particular attention to SPF, Sender ID, and DomainKeys.

Who are you?
the overriding question

How can I tell you are who you say you are?
authentication

Why should I accept your mail?
receiver policy

– Everybody knows I'm a good guy.
reputation

– Because Bob says I'm a good guy.
accreditation

Audience

If you are a domain owner, systems manager, DNS administrator, email administrator, or brand manager, you should read this paper.

How to Read this White Paper

Part I, ***A Vision for Spam-Free Email***, offers a birds-eye-view of the antispam landscape and lays out a strategy for reaching a spam-free world. It walks through an end-to-end message delivery scenario and shows how authentication, reputation, and accreditation fit together.

Part II, ***About Sender Authentication***, explains the differences between IP-based and crypto-based authentication and discusses the most promising technologies.

Part III, ***Deployment***, explains what email senders, receivers, ISPs, MTA vendors, MUA vendors, and volume ESPs need to do, and suggests a schedule for coordinated rollout.

A Vision for Spam-Free Email

Content filtering is reaching the end of the road. The Aspen Framework will take its place. If you are familiar with the Accountable Net concepts developed at the Aspen Institute in December 2003 and with the three-part model of authentication, reputation, and accreditation, you can skip directly to the next section, "About Sender Authentication."

http://www.aspeninstitute.org/Programt3.asp?bid=13218
http://www.edventure.com/conversation/article.cfm?Counter=367986

The Problem of Abuse

Anyone can send email to anyone else, within seconds, at zero apparent cost. That is the greatest strength of the Internet mail system. It is also its greatest weakness. Because the system is biased in favour of delivery, it is prone to abuse in the form of spam, viruses, and phishing scams. The very features that made email successful now threaten its viability.

To combat abuse we must add accountability. On a social level, legislative approaches such as CAN-SPAM attempt to punish spammers for their trespasses. On a technical level, sender authentication protocols give computers new ways to automatically distinguish forgeries from authentic messages.

The Underlying Concept

If you step back and squint, every plan for solving spam looks roughly the same.

Senders are asked to do X. Receivers are asked to check for X. If X is missing, receivers are to assume the message is spam. X is meant to be hard for spammers and easy for good guys.

Approaches mostly differ about what exactly goes into X. The challenge is to make X as lightweight as possible while remaining robust and secure.

Under the Aspen Framework, X is two things together: authentication and reputation. (The third thing, accreditation, is used as a buffer for when the reputation part fails.)

Authentication, in turn, enjoys a surfeit of competing and complementary technologies.

The vision described here is actually an amalgam of three related visions: the gospel according to Sender ID, the gospel according to SPF, and the gospel according to DomainKeys.

Drivers; or, Who's Buying It

Receivers who want to reduce costs and improve their user experience are expected to embrace the vision. Senders who want to improve deliverability are also expected to go along. Between the two, network effects will drive both senders and receivers in the direction of sender authentication.

Reversals from the Paradigm Shift

The opposite of every great idea is another great idea. –Niels Bohr

mengwong@pobox.com 20040920

In the 21st century, if a message is not from an accountable sender, it should expect to be rejected.
Senders must be authenticated. Senders must also be known, reputable, or accredited.

20th century email		**21st century email**
The average message is good. Spam is the exception.	1	*The average message is spam. Ham is the exception.*
By default, accept a message unless we have a good reason to reject it.	2	*By default, reject a message unless we have a good reason to accept it.*
Spammers evolve. The list of reasons to reject a message keeps growing.	3	*Good senders are relatively static. The list of reasons to accept a message stays short.*
Filter out spam based on content.	4	*Filter in ham based on sender.*
File suspected spam to a spam folder.	5	*There is no spam folder.*
Spamfolders reduce reliability. Senders have to ask "did you get my mail?"	6	*If a message is accepted, senders can be confident it will be read.*
The biggest challenge in solving spam is reducing false positives.	7	*If we can solve false positives perfectly, spam is solved as a side effect.*
End-users can send mail through any SMTP server, as anyone.	8	*End-users have to phone home using 587 AUTH and send mail as themselves.*
Expectation: strangers can email each other totally out of the blue.	9	*Expectation: strangers need to be generally reputable or else be introduced.*
Corporations, particularly sales accounts, are very sensitive to FPs, so the "default accept" paradigm will never go away entirely.	10	*Humans, particularly children, are much more sensitive to false negatives, so "default reject" will eventually become dominant.*

Vision Walkthrough

> *… All experience hath shewn, that mankind are more disposed to suffer, while evils are sufferable, than to right themselves by abolishing the forms to which they are accustomed. […] it is their right, it is their duty, to throw off such Government, and to provide new Guards for their future security.*
>
> Declaration of Independence

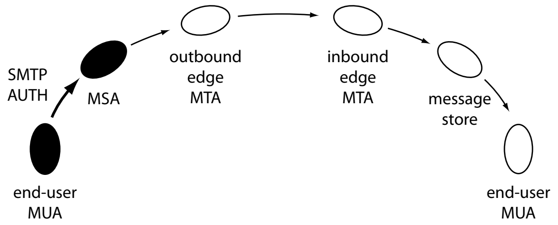
The vision for a spam-free email system contains many parts. This walkthrough describes the life cycle of an email message under a strong version of the vision. In practice, it may be possible for you to alter or selectively weaken certain parts of the model to suit local conditions.

An *MUA* submits a message to an *MSA* using *SMTP AUTH.*

At present, many Mail User Agents (MUAs) are configured to submit messages to a Mail Submission Agent (MSA) over port 25. The MSA accepts the message because the IP address of the MUA is trusted.

The vision calls for MUAs to authenticate themselves with a username and password to the MSA over port 587. Armed with that username, the MSA can better implement outbound anti-abuse policies such as rate limiting. An audit trail is also easier to follow. VPN submission is a good alternative.

MUA: (n) Mail User Agent. What the end-user thinks of as "my email program" and what ISPs think of as "the email client". Popular MUAs include Eudora, the Outlook family, and Mac Mail. Determines what the end-user sees of an email message. The entity responsible for signing and verifying S/MIME cryptographic authentication. Possibly also the entity responsible for verifying Sender ID (PRA) authentication.



An accountable sender has published *authorization records* in DNS.

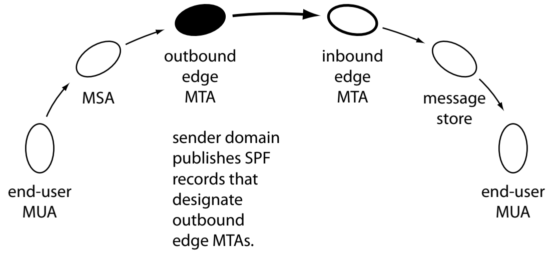
At present, any host on the net can claim to be a Mail Transfer Agent (MTA). Open relays, open proxies, and zombies are hard to distinguish from "official" MTAs run by responsible entities. Numerous DNS Block Lists (DNSBLs) attempt to identify the offenders, but they are inherently limited to a reactive mode of operation. Furthermore, inaccurate listings often cause "collateral damage," and they can be hard to correct. Finally, playing "whack-a-mole" with 4.3 billion IP addresses is a difficult scaling problem. On the flip side, ISPs may maintain lists of IP addresses of known good senders. Maintaining those lists is a time-consuming endeavour.

The vision calls for accountable participants in the email system to • authorize certain MTAs as designated senders, • add cryptographic signatures to outgoing messages, or • do both. IP-based authentication lets receiver MTAs distinguish accountable MTAs from hijacked machines. Crypto-based authentication lets receivers authenticate message content without reference to the sending MTA. The two approaches are complementary and reinforce each other. Both involve publishing records in DNS.

MTA: *(n)* Message Transfer Agent. What ISPs think of as "the email server" and what end-users often don't think of at all. Popular opensource MTAs include Sendmail, Postfix, Qmail, and Exim. Popular commercial MTA vendors include Sendmail, Openwave, Ironport, Microsoft Exchange, and others. Can operate as a sender or as a receiver. When receiving, responsible for verifying SPF and other authentication.

MSA: *(n)* Message Submission Agent. What an MUA thinks of as an "SMTP server". Usually set up by an ISP to receive mail from end-users. Often whitelists the dialup/broadband range. May also be the outbound edge MTA. Unlike a receiving MTA, must require SMTP AUTH when accepting connections from outside trusted network. See RFC2476.

Zombie: *(n)* An end-user machine under the control of a virus, often used to send spam. It is estimated that up to one in three machines might be infected with worms and viruses. Zombies can send spam direct-to-MX or routed through an ISP MSA.



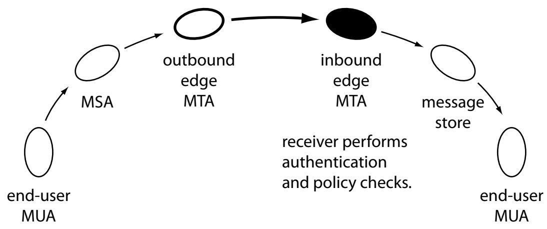
An *authorized outbound edge MTA* transfers a message to an *inbound edge MTA.*

Edge MTA: *(n)* On the sending side, an MTA which takes the role of an SMTP client to the public Internet. On the receiving side, a server MTA which accepts connections from the public Internet from sending MTAs.

It takes a fair amount of expertise for a human to extract the SMTP client from "Received" headers and guess whether it is authorized to send mail on behalf of the purported sender. At present, there is no easy way for computers to do the same thing.

Also, while it is possible to say that a given message, if signed, is authentic, it is not currently possible for computers to conclude that a message not signed is a forgery.

The vision calls for receiving MTAs to automatically verify incoming SMTP sessions against the information published in DNS by senders. This constitutes the sender authentication step. There are three main classes of results: PASS, FAIL, and UNKNOWN.



The receiving MTA also makes a *receiver policy decision* about senders.

All senders can authenticate themselves. Spammers will too. Therefore authentication checks alone are not sufficient.

The vision calls for the receiving MTA to also decide if it likes or dislikes the sender. This decision can be based on a purely local opinion. It can also be informed by opinions from third parties.

Today, many ISPs keep local whitelists and blacklists. These lists are not shared with their peers, and so their utility is limited. Many ISPs also use public whitelists and blacklists.

The end-user addressbook can be an input to that decision.

If you are in my addressbook, I would probably be happy to read mail from you.

The vision calls for the receiver's MUA to help distinguish wanted mail that passes authentication from unwanted mail that also passes authentication. If the MTA happens to know what's in the end-user's addressbook, then that decision can be optimized up into SMTP time.

In fact, if you're in my best friend's addressbook, I would probably be happy to read mail from you. Emerging technologies such as LOAF (http://loaf.cantbedone.org) and http://www.web-o-trust.org/ are good examples of experiments in this space.

***Reputation and accreditation services* record assertions about senders.**

At present, reputation services exist in the form of DNSBLs. They are based on IP address and generally assert an opinion that a given IP address should be blocked. Accreditation services also exist in the form of DNSWLs. They vouch for assertions made by legitimate senders who care very much about deliverability. They may be backed by some kind of financial bond.

The vision calls for these services to also • keep track of senders by domain name and to • indicate if, and why, certain domains should be considered good. This makes it possible for receivers to recognize known good senders and confer a sort of "first-class" status to their messages. Messages that pass the joint test of authentication and reputation can then choose to bypass other more expensive and potentially error-prone content-based antispam tests.

A useful review of many blacklists can be found at http://www.sdsc.edu/~jeff/spam/Blacklists_Compared.html

Reputation service:* *(n) offers opinions about senders, usually based on empirical observation of *past behaviour*. Operates on behalf of receivers. May be expressed as spamtrap counts, a binary vote, a ratio of complaints to total message volume, etc. Receivers need to decide for themselves what opinions mean.
Examples: movie reviews, DNSBLS (Spamhaus, Spamcop).

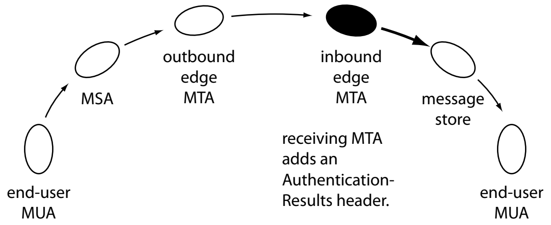
Accreditation service:* *(n) offers assertions about senders, usually based on representations made by senders. Operates on behalf of senders. May be expressed as a list of reasons for predicting *future behaviour*.
Example: "rated R", Bonded Sender, Habeas, Verisign VDL.

How to tell the difference? Very crudely: follow the money! If the sender pays to be listed, it's accreditation. If the receiver pays them for their opinions, it's reputation.

The receiving MTA records the results of the joint tests in the message headers.

At present, MTAs do not generally record the results of DNSBL lookups in a user-visible form. If a sender is found on a DNSBL, the message is simply blocked or dropped. Otherwise it is let through. The reasons for accepting or rejecting a message are generally not exposed to end-users.

The vision calls for receiving MTAs to record attach authentication and reputation metadata to messages. The most natural place to put that data is in the headers. Spammers will try to spoof those headers, but that problem can be solved because communications between a receiver MTA, a message store, and an end-user MUA occur within a confined space.



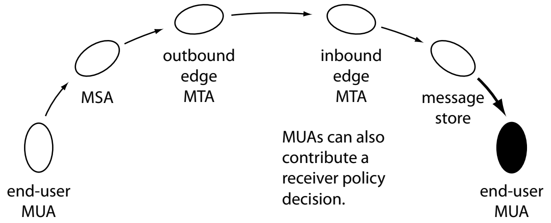
receiving MTA adds an Authentication-Results header.

The receiving end-user's MUA displays a confidence mark.

At present, MUAs lack a consistent, industrywide, visual language for describing the confidence an end-user should place in a message. We need the equivalent of the HTTPS padlock icon.

The vision calls for receiver MUAs to add an explanatory visual element to displayed messages. That element can help fight forgeries by warning of authentication FAILS. This helps combat phishing and spoofing. MUAs should also distinguish a dual-PASS result (where both authentication and receiver policy tests approve the sender) to help fight false positives and improve deliverability of legitimate mail.

When the MUA is actively involved in analyzing and classifying the message, it has all the information it needs to do this. When the MUA is operating passively downstream from the point where that decision is made, it can simply display the information recorded in the message headers.



MUAs can also contribute a receiver policy decision.

"Not Junk": The opposite of the Junk Mail folder.

At present, MUAs may file suspected spams into a Junk Mail folder based on Bayesian filtering and other logic.

The vision calls for MUAs to file dual-PASS messages into the semantic opposite of the Junk Mail folder – "spamproof", "first-class", or "Not Junk".

At present, the default user expectation for an email inbox is that it will accept all messages by default, unless a set of complex heuristics intervenes and identifies some messages as spam.

The vision calls for the market to move toward a default-reject orientation. Once the "Not Junk" folder gains wide acceptance, the next step is to think about simply rejecting any messages that would not make it into that folder.

It is rumoured that 10 to 15% of AOL and Earthlink's userbase have switched to whitelisting-only spam filtering.

IP-based authentication validates the channel that transported the message, and tends to focus on the sender. Crypto-based authentication focuses on the original author.

An Example

Google Mail (`gmail.com`) is an early and enthusiastic adopter of sender authentication technologies. They publish SPF records and sign messages with DomainKeys. Here are some headers from a message they sent from mengwong@gmail to mengwong@dumbo.pobox.com:

Sendmail, Inc is another enthusiastic pioneer in sender authentication. See http://www.sendmail.net/ for details on the Messaging Integrity Pilot Program.

```
Delivered-To: mengwong@dumbo.pobox.com
Received-SPF: pass (dumbo.pobox.com: domain of mengwong@gmail.com designates 64.233.170.199 as permitted sender)
Received: from rproxy.gmail.com (rproxy.gmail.com [64.233.170.199])
        by dumbo.pobox.com (Postfix) with ESMTP id 9C02E198
        for <mengwong@dumbo.pobox.com>; Wed, 27 Oct 2004 23:09:16 -0400 (EDT)
Received: by rproxy.gmail.com with SMTP id 80so309rnk
        for <mengwong@dumbo.pobox.com>; Wed, 27 Oct 2004 20:09:12 -0700 (PDT)
DomainKey-Signature: a=rsa-sha1; q=dns; c=nofws;
        s=beta; d=gmail.com;
        h=received:message-id:date:from:reply-to:to:subject:mime-version:content-type:content-transfer-encoding;
        b=Vlj/++WbtRXfAdBGd+9GjE2ggK8e5Fwe2H68kpHOh7yFu9NHrRwjAeWpcar84+s+UWEsTWLLBdwGnabOfLeGOlOLSdxeUbrQ4ibPO
          QUOF10ZkalycNmrpG3tIvuE5ta9w1+kLEwJs1d7PJU24XyBsqp+mdyMWT6mroXi0GXzBps=
Received: by 10.38.98.18 with SMTP id v18mr773189rnb;
        Wed, 27 Oct 2004 20:09:12 -0700 (PDT)
Received: by 10.38.8.32 with HTTP; Wed, 27 Oct 2004 20:09:12 -0700 (PDT)
```

Note the `Received-SPF` and `DomainKey-Signature` lines.

History

In 1998 Jim Miller had the idea of designating outbound mailers. In 2002 Paul Vixie wrote up the idea in a paper titled “Repudiating Mail-From”. In 2003 Hadmut Danisch independently authored a specification called “Reverse MX” (RMX). Around the same time, Gordon Fecyk wrote a similar specification called “Designated Mailer Protocol” (DMP). Later that year Meng Weng Wong combined some features of RMX and DMP into SPF. SPF originally stood for Sender Permitted From, but changed its name to Sender Policy Framework. Microsoft® also wrote a specification, “Caller-ID for Email” (CID), as part of a Coordinated Spam Reduction Initiative proposal (CSRI). While they differed in a number of ways, all of these proposals shared the concept of using DNS records to authorize SMTP clients to be MTAs.

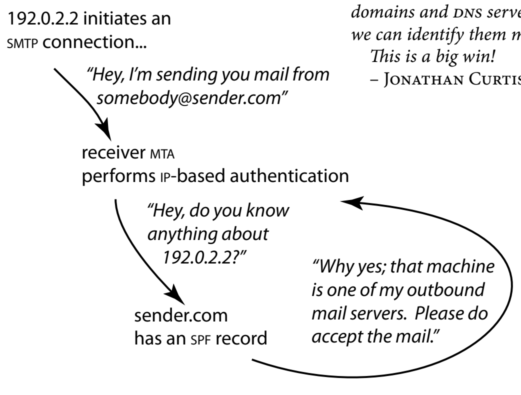
While all this was going on, Yahoo!® was developing a crypto-based approach, called DomainKeys (DK). In early 2004 rough consensus appeared in the email industry to proceed with both IP-based and crypto-based approaches. To make things easier, Microsoft® and Meng worked to merge CID and SPF into a single proposal in the MARID working group of the IETF. Work began in earnest in May and a converged specification was concluded in October under the name Sender ID™.

There was another bookish lad in the town, John Collins by name, with whom I was intimately acquainted. We sometimes disputed, and very fond we were of argument, and very desirous of confuting one another, which disputatious turn, by the way, is apt to become a very bad habit, making people often extremely disagreeable in company by the contradiction that is necessary to bring it into practice; and thence, besides souring and spoiling the conversation, is productive of disgusts and, perhaps enmities where you may have occasion for friendship. I had caught it by reading my father's books of dispute about religion. Persons of good sense, I have since observed, seldom fall into it, except lawyers, university men, and men of all sorts that have been bred at Edinborough.

Autobiography
Benjamin Franklin

How IP-based Authentication Works

Most legitimate mail from a given domain enters the Internet from a relatively small set of servers. If we can identify those servers and list them in machine-readable form in DNS, receivers can easily check incoming messages against that list. Messages that come from an approved server are considered authenticated. Messages that don't come from an approved server may be considered forgeries. Exactly how a receiver treats suspected forgeries depends partly on what the sender has specified as a default in such cases.


192.0.2.2 initiates an SMTP connection...
"Hey, I'm sending you mail from somebody@sender.com"
receiver MTA performs IP-based authentication
"Hey, do you know anything about 192.0.2.2?"
sender.com has an SPF record
"Why yes; that machine is one of my outbound mail servers. Please do accept the mail."

This forces spammers to have their own domains and DNS servers, which means we can identify them more easily.
This is a big win!
– JONATHAN CURTIS, Georgia Voter

The SPF record

SPF records use a simple syntax which any DNS administrator should find intuitively familiar. Records are easy to set up by hand. Here is an example record: `v=spf1 mx`

It means that the MX servers for the domain are explicitly permitted to send mail from that domain.

SPF features a "best-guess" technology which basically says: if a domain does not publish SPF records, try "`a/24 mx/24 ptr`" anyway. If that returns a PASS, consider that PASS a useful first approximation. This technique significantly reduces the deployment burden for technologically unsavvy senders who are lucky enough to obtain a PASS using best-guess alone.

SPF Check: *(n)* a test of the validity of an IP address against a domain name. If the domain name comes from the return path, you're doing an SPF Classic or SPFv1 check. If it comes from the PRA, you're doing a PRA check.

How SPF Classic Works

Every SMTP transaction begins with a MAIL FROM command. SPF Classic examines the return-path identity given in the MAIL FROM command. It tests the client IP against the SPF record for the domain in the return-path. As it focuses on the return-path, SPF Classic is most obviously useful for helping fight bounce floods caused by undeliverable forgeries. But it is also useful in cases where the recipient of a forgery is deliverable. Even though the forgery may not result in a bounce, some harm can still be prevented.

If the MAIL FROM is empty, SPF Classic falls back to the HELO argument. This is useful for handling bounce scenarios and closes a loophole whereby spammers could just send mail with a null MAIL FROM.

SPF record: *(n)* a `v=spf1` record. `spf2.0` records are for special cases only. (Caller-ID records using the XML format are no longer used.)

Bottom Line: The SPF records that you create need to work in both MAIL FROM and HELO contexts. Someone might be looking at your SPF record because your domain showed up in a MAIL FROM, or because the MAIL FROM was blank and your domain showed up in the HELO. If none of your servers HELOS using a given domain name, then you only have to worry about MAIL FROM use of that name. If you're setting up an SPF record for a hostname, you probably do want to ensure the record will work for HELO. You can do this easily by adding an "`a`" mechanism.

How Sender ID works

When an MUA displays a message, it shows who sent the message, usually by extracting the `From:` header. Every well-formed email message contains a `From:` header. But a message may also contain a `Sender:` header. In that case, an MUA may say to the end-user "From *Sender* on behalf of *From*". Microsoft took this concept one step further and defined the Purported Responsible Address (PRA). A Sender ID compliant MUA displays: "From *PRA* on behalf of *From*". Sender ID as originally conceived runs an SPF test, but uses the PRA instead of the MAIL FROM.

The PRA is drawn not just from the `From` and `Sender` headers, but from the `Resent-From` and `Resent-Sender` headers as well. See the Sender ID specification for details.

Sender ID was recently resubmitted to the IETF. It now specifies that both the return-path and the PRA may be used. Software that implements only SPF Classic can therefore be called Sender ID compliant. In practice, most people associate Sender ID with the PRA, and SPF Classic with the MAIL FROM, or return-path.

The author recommends that MUA software implement Sender ID with PRA checking. The author recommends that MTA software implement Sender ID with MAIL FROM checking, aka SPF Classic. The author also recommends watching to see how crypto develops.

Politics. Some have characterized Sender ID as "SPF with PRA bolted on." The IETF community has roundly criticized the PRA on both technical and licensing grounds. Some members of the technical community assert that the very idea of using PRA for header validation is flawed. Other observers comment that the patent license which surrounds PRA makes it unpalatable to implement.

A potential security vulnerability may occur if an MTA validates a PRA address which is not actually displayed to the end-user. If an upstream MTA validates the PRA and records the authentication results in a header, and if an MUA displays that authentication result as a mark of confidence, that MUA must be very careful to also display the validated address to the end-user. Until more MUAS display the PRA, the author expects few MTA software engines to implement PRA checking. Commercial MUA software, however, will probably find it useful, at least until cryptographic solutions designed expressly for 2822 content checking mature.

At least one major commercial MUA vendor has publicly stated it will proceed with PRA checking. Microsoft appears to be preparing to turn on Sender ID checking in Hotmail, Outlook, and Exchange. The latest versions of Outlook are expected to display the PRA where available, rather than just the Sender header.

Sendmail, Inc. has released experimental milters for Sender-ID that check both the MAIL FROM (SPF Classic) and the PRA. Other commercial MTA vendors have done the same. Generally, however, SPF Classic is more widely supported than PRA at this time.

The Patent Situation. Microsoft has two patents pending on Sender ID. While the patent license offers Royalty Free terms, according to Lawrence Rosen Esq., the sublicensing provision of the patent license makes PRA incompatible with GPL free software such as Exim. Apache SpamAssassin, for example, have stated that they will not implement PRA checking, and will stick to SPF Classic in version 3.0 and above. Note that the patents only cover the PRA and do not restrict SPF Classic in any way. Some individuals are planning to work with http://www.pubpat.org/ to challenge the patent on grounds of prior art and obviousness.

How Cryptographic Techniques Work

All cryptographic approaches to authentication agree on the basic concept: sign some portion of the message content and present that signature for verification. The approaches disagree about what gets signed, where the signature goes, and how verification is done.

PGP and GPG sign the message body only and put the signature directly in the body. Keys are stored in end-user keyrings or in public keyservers; key management uses a peer-to-peer web-of-trust architecture. The signature includes a description of the signing entity, but MUAs tend not to use that author data from the signature to override the `From:` header.

S/MIME signs the message body also, but presents the signature in a logically distinct MIME part. Keys are signed by a certificate authority, so key management follows a hierarchical model similar to SSL. Signatures that do not match the `From:` header tend to result in some sort of MUA user interface warning.

DomainKeys, originally championed by Yahoo!**,** signs the body and some message headers. It puts the signature in a `DomainKey-Signature` header. Keys can be self-signed, as in PGP, and published in DNS following a decentralized, opportunistic encryption model. If a message fails signature verification, it should be rejected by the receiving MTA during SMTP time, but in practice will probably result in some sort of warning sign in the MUA.

BATV signs the return-path only and places the signature in the return-path. If a sender always BATV signs its return paths, any bounces that come to a non-BATV-signed address must be bogus. Though there is provision for a public mode, BATV primarily uses a private-secret key scheme because only the signing system absolutely needs to authenticate its signatures. BATV is a lot like VERP, but with a signature.

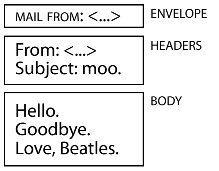
The latest evolution of SES also places the signature in the return-path, but signs the message headers and body as well, much like DomainKeys. SES also uses a private-secret key scheme, but validation can occur at SMTP time! How? The sender system publishes an `exists` mechanism using SPF; that mechanism instructs receivers to include the full localpart in a DNS query against the sender. When the sender's DNS server gets a query, it validates the signature. If a sender system signs all outgoing mail with SES and runs a custom SES-enabled DNS server which can validate localpart queries, it no longer needs to worry about forwarding false positives.

Jim Fenton's Identified Internet Mail is similar to DK.

Microsoft's Email Postmarks is essentially MTA-to-MTA S/MIME which may be easier to implement in Exchange.

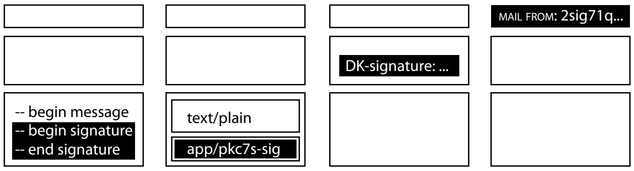
The IETF MASS SIG has engaged the above proposals.

What should the signature cover, and where should it go? Different technologies answer the question differently.



RFC2821 describes the envelope.
RFC2822 describes message headers and body.

The Evolution Of Cryptographic Signatures in Sender Authentication



The earliest incarnations of PGP ran the signature directly inside the body text.

When S/MIME was invented, signatures got their own MIME part.

DomainKeys and IIM move the signature into the headers.

SES and BATV move the signature all the way back into the MAIL FROM return-path, like VERP.

PGP and S/MIME have been around for a long time, but they are not widely used. While most MUAs today support S/MIME natively, the vast majority of mail sent is not signed. Why not? I don't know. Maybe the average end-user doesn't sign their outbound mail because they find it inconvenient to manage keys and enter passphrases. But why don't the bulk mailers, even the well-organized volume ESPs, sign their mail? Maybe they perceive a significant population of MUAs that don't support S/MIME, and fear customer complaints. I am not aware of any published results that give a basis for this fear.

In 1999, the US Department of Defense published a policy mandating future use of PKI technologies. They chose S/MIME over PGP because they rejected the PGP web of trust model in preference for the hierarchical Certificate Authority model. Use of PKI certificates to access DoD restricted access web sites became mandatory October 1, 2004. The DoD may similarly mandate S/MIME for email in the future.

See http://mipassoc.org/batv

Mailing lists use Variable Envelope Return Path to track bouncing subscribers.

See http://ses.codeshare.ca/

Bottom line: cryptographic signing is performed by the MTA. In IP-based protocols, senders have to go fiddle with DNS. With crypto, senders have to put their public keys in DNS, and also have to upgrade to an MTA that supports signing.

IIM repeats the signed headers within the headers. This is better for mailing lists. It also includes the public key with every message. DK and IIM are similar enough that many industry insiders hope and expect them to merge in early 2005.

See http://www.imc.org/ietf-mailsig/index.html and
http://www.elan.net/~william/emailsecurity/emailsignatures-comparisonmatrix.htm

Using Multiple Approaches

The Achilles' Heel of IP-based schemes is forwarding: unless the forwarder rewrites the return-path, the final receiver will consider the message a forgery because it doesn't come directly from the original sender.

The Achilles' Heel of cryptographic schemes is content munging: many mailing lists alter the message content during transit. Unless the mailing list manager re-signs the message, the final receiver may consider the message a forgery because the content has changed since the signature was created.

"To unsubscribe, send a message to ..."

SPF Classic, an IP-based scheme, works well with mailing lists, because mailing lists always change the return-path to be the mailing list bounce handler. DomainKeys, a crypto-based scheme, works well with forwarding, because forwarding doesn't usually change message content.

Because one scheme's meat is another scheme's poison, it seems only prudent to use multiple schemes: that way, the strengths cancel out the weaknesses. This is the basic idea behind Unified SPF. Unified SPF is a syncretist theory that also admits HELO, IP, PTR, and PRA checking.

CSV is one proposal that focuses on the HELO or EHLO hostname argument as the identity for verification.

The author recommends that prudent senders do at least two things: they should publish SPF records and eventually sign messages with DomainKeys. Prudent receiver ISPs should check SPF Classic and DomainKeys at the MTA.

Verbatim forwarding

…is a common practice: anyone who has seen an `/etc/aliases` or `.forward` file knows what it is. Messages sent to a forwarding alias are remailed to the final destination. Unfortunately, those messages are often reinjected without any indication that forwarding occurred: the return-path remains unchanged, and no special headers are added to the message.

Forwarding is a problem for SPF Classic and Sender ID because forwarded messages are difficult to distinguish from forgeries! Under SPF Classic, forwarders can help by implementing SRS – Sender Rewriting Scheme. Under Sender ID, forwarders are expected to prepend a `Resent-From` header. Neither is convenient for forwarders. Receivers can also help by simply whitelisting known forwarders by IP address or PTR name.

Forwarding is better under DomainKeys because forwarding generally doesn't munge content.

Bottom line: good for DK, bad for SPF.

Mailing lists

…tend to append text to the body of a message. This is a problem for DomainKeys because content munging invalidates the signature. DomainKeys expects mailing list managers (MLMs) to re-sign messages and claim authorship after any content munging. This is not convenient for mailing lists.

Mailing lists reset the return-path so that bounces go back to the MLM, not the original author. This is good for SPF Classic.

While any mailing list worthy of the name resets the return-path, only some mailing lists add a Sender: header. EzMLM, a widely-used MLM, does not. Yahoo!Groups doesn't add Sender: either, though in theory that can be fixed easily … if Yahoo! cooperates.

Sender ID asks all mailing lists to add a `Sender:` header. This is not convenient for the installed base of mailing lists who do not add `Sender:`.

Bottom line: good for SPF, bad for DK.

So let's do DK … and let's do SPF!
Imagine this: `v=spf1 a mx dk -all`

Reputation Systems

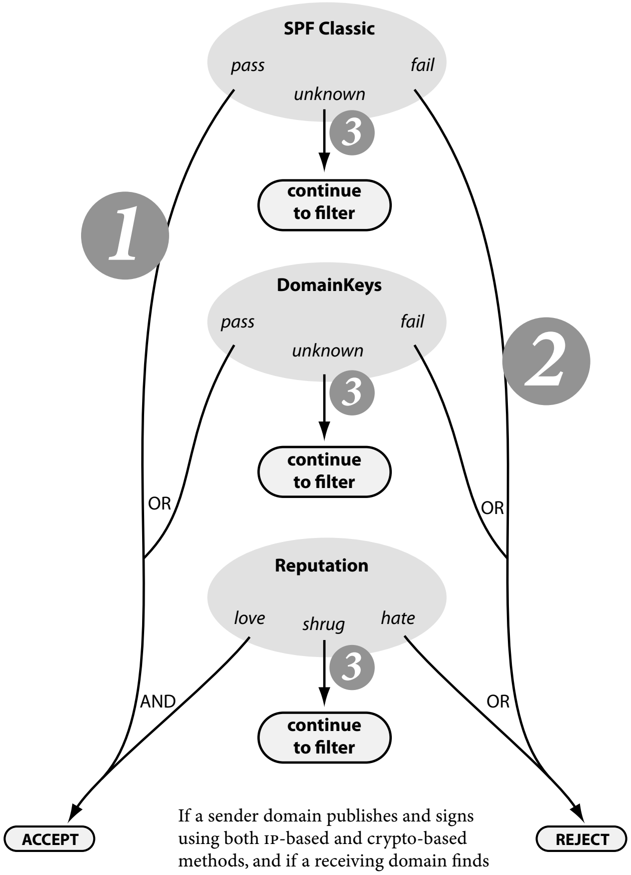
Authentication alone is not enough. Reputation systems are a key component of the Aspen Framework. They help receivers decide if a mail from an authenticated sender is desirable or undesirable. Just as banks rely on credit rating agencies, receivers rely on reputation systems.

A detailed discussion of reputation systems is outside the scope of this white paper. But here is the main point: first-generation reputation systems tried to enumerate all the IP addresses which they considered bad. Next-generation reputations will try to enumerate all the domain names which they consider good. There are many ways to do this. All of them are interesting.

What if a domain has no reputation? If it is patient, it can gradually acquire one simply by sending enough good mail to be noticed over time. After all, sensible receivers should not reject outright mail from senders without a reputation; receivers might just graylist or filter more aggressively. But if the domain is impatient or wishes to refute a bad reputation, it may choose to sign up with an accreditation service.

The Karma Project is an aggregation service which reduces the burden on receivers. Instead of talking to N services to get N results, receivers can talk to a single service to get N results. Please contact the author for details on feeding into and reading from the Karma system.

BondedSender.com and Habeas are examples of services which vouch on behalf of senders.



If a sender domain publishes and signs using both IP-based and crypto-based methods, and if a receiving domain finds that neither method passes, it can be very confident that the message is a forgery.

Deployment: For Email Senders

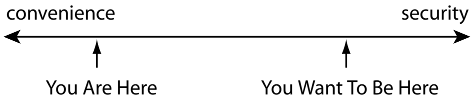
All senders and receivers of email — from the largest ISP to the smallest personal domain — should deploy one or more forms of sender authentication in 2005. You can deploy SPF and Sender ID today. This section tells you how. Comments on DomainKeys are also provided for comparison.

In this example, you are `sender.com`.

Under DomainKeys: you need to do different things for DomainKeys. These sidenotes briefly describe what to expect under DK. DK is still under development and may incorporate features from IIM. When the crypto approaches settle, expect a future version of this whitepaper to include detailed deployment instructions. Meanwhile, keep in mind that when people say DomainKeys, they may mean a future, more mature version.

First, prepare.

Stopping forgery means stopping *all* forgery: good and bad. Over the years people may have gotten used to the lax nature of email, and your security-minded efforts to close loopholes may encounter resistance from some people who find those loopholes quite comfortable and you quite annoying. You are not alone: this is a classic change management problem. Sooner or later every computer professional encounters the eternal tension between security and convenience. If you want to prevent bad guys from forging your identity, and if you want to do it by limiting the approved routes for outbound mail, your end-users are going to have to use those routes or risk being classified as one of the bad guys.



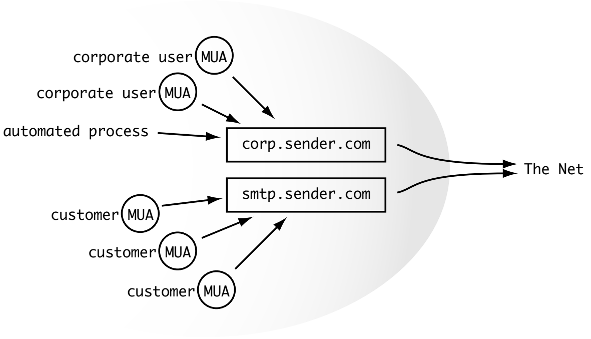
Audit Your Outbound Mailstreams

Under DomainKeys: you need to audit your outbound mailstreams too.

To begin, you need to identify all the ways in which legitimate mail from sender.com goes out onto the Internet. Outbound mail usually comes from two sources: humans and machines.

Mail from humans. Suppose end-users with addresses like `username@sender.com` set the SMTP server in their MUA to `smtp.sender.com`. That is one mailstream: you need to find out how mail submitted to `smtp.sender.com` appears to the outside world. In the simple case, `smtp.sender.com` has public DNS records that identify its outbound IP addresses.

Mail from machines. Machine-generated processes may also originate mail from addresses at `sender.com` – for example, `automated-billing@sender.com`. You need to identify all such processes and the servers through which they inject messages into the Internet. In the simple case, a corporate system `corp.sender.com` may be responsible for customer service and automated billing.



Under DomainKeys: you upgrade your MTAs to sign all outbound mail, and you publish public keys in DNS. Upgrading MTAs is generally regarded to be more work. Still, this approach is worth the effort.

Construct the record

You've enumerated all the servers that originate mail from `sender.com`. Now you're ready to describe them in SPF syntax. A number of wizards are available online to help you create your SPF record. If you have a complex configuration, you should see the SPF protocol specification for full details.

If both corp and `smtp.sender.com` send mail that looks like this:

```
MAIL FROM:<somebody@sender.com>
From: Some Sender <somebody@sender.com>
```

Assuming the following existing DNS entries:

```
sender.com       A    192.0.2.222
sender.com      MX 10 smtp.sender.com
smtp.sender.com  A    192.0.2.2
corp.sender.com  A    192.0.2.222
```

Here are some examples of how sender.com's SPF record might look:

```
sender.com TXT “v=spf1 a:corp.sender.com a:smtp.sender.com ?all”
sender.com TXT “v=spf1 a mx ~all”
sender.com TXT “v=spf1 ip4:192.0.2.0/24 -all”
```

Think briefly about PRA and Mail-From contexts.

You've identified and described the servers that send mail from `sender.com`. But there are actually two identities involved: the MAIL FROM return path in the RFC2821 envelope, and the PRA identity extracted from the RFC2822 headers. In the vast majority of cases, when you compose a mail message, the return-path and the PRA are the same, and you only need to create a single `v=spf1` record. But if your situation is complex and you routinely create messages whose MAIL FROM and PRA differ, you may need to create an `spf2.0/pra` record as well. In that case, your record might look like:

```
v=spf1 a:corp.sender.com a:smtp.sender.com ~all
spf2.0/pra ip4:192.0.2.0/24 ~all
```

Together, these records mean:

- if the SMTP client is `corp.sender.com`, then `sender.com` may legitimately appear in the return-path.
- if the SMTP client is `smtp.sender.com`, then `sender.com` may legitimately appear in the return-path.
- if the SMTP client is anything else, `sender.com` should not appear in the return-path.
- if the SMTP client is in the `192.0.2.0/24` subnet, then `sender.com` may legitimately appear in the PRA headers.
- Otherwise, `sender.com` should not appear in the PRA headers.

Bottom Line: Unless you're an outsourced Email Service Provider, you probably don't need to worry about customizing your record for the PRA. Getting it right for MAIL FROM should be your first concern. But if for whatever reason you want to explicitly disable PRA checks, just add a blank record: `sender.com TXT "spf2.0/pra"`

PRA: The Purported Responsible Address is used primarily by Microsoft MUA software. Opensource and MTA software generally prefer to examine the MAIL FROM return path.

FAQ: *Do I need to publish an* `spf2.0/pra` *record if it contains the same data as my* `v=spf1` *record?* No. Software that operates in PRA context will grok `v=spf1` records just fine. Use `spf2.0/pra` only to override.

Under DomainKeys: DK deals only with 2822 information, and with only the From: and Sender: headers. It leaves 2821 validation to SPF. One might say that DomainKeys attempts to validate authorship, and SPF Classic attempts to validate the last hop sender.

Test the record, part 1

Figuring out your SPF record is the first step, but how do you know it's doing what you want? There are a number of SPF validation tools on the web. You paste your proposed SPF record into the tool, and it tells you whether it's syntactically correct.

Some validation tools are listed at `http://spf.pobox.com/certification.html`

Put the record in DNS

SPF records are published in DNS as TXT records. There are two common scenarios: domain hosting and direct control.

If your domain is hosted, your hosting provider should offer a web interface. Most hosting providers have started offering a TXT option so customers can publish SPF records. If your domain hosting provider does not, you may want to transfer management of your domain to another provider, or petition them to add support for TXT.

If you run the nameservers for your domain, you ought to be familiar with editing zone files. Common nameservers include BIND and djb's tinydns. An SPF record in a BIND zonefile might look like this:

```
sender.com. IN TXT "v=spf1 ip4:192.0.2.0/24 a mx ~all"
```

The equivalent record in tinydns might look like this:

```
'sender.com:v=spf1 ip4\072192.0.2.0/24 a mx ~all:300
```

You can confirm that the record appears in DNS by using `nslookup` or `dig`. Until you're comfortable with the record, you should keep the TTL low, so you can change it quickly if you need to.

A new Resource Record type is being assigned, but adoption of new RR types is generally held to be a slow process. So SPF records make do with TXT, which is widely supported and not explicitly reserved for anything else.

A list of hosting providers which support TXT records can be found at `http://www.spf.idimo.com/txt-supporters.html`

There is no central registry for publishing SPF information. Your SPF record is published directly in the DNS for your domain, which ultimately you control.

Under DomainKeys: messages acquire an additional header that signs the message content. For the example on p 9, the corresponding public key appears in DNS:

```
beta._domainkey.gmail.com. IN TXT
"t=y; k=rsa; p=MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQC69TURXN3oNfz+G/m
3g5rt4P6nsKmVgU1D6cw2X6BnxKJNlQKm10f8tMx6P6bN7juTR1BeD8ubaGqtzm2rWK4Li
MJqhoQcwQziGbK1zp/MkdXZEWMCflLY6oUITrivK7JNOLXtZbdxJG2y/
RAHGswKKyVhSP9niRsZF/IBr5p8uQIDAQAB"
```

Test the record, part 2

Getting your SPF record into DNS is a big step. Now everybody can read it! Again, you should use a validation tool to check it. But this time, instead of pasting your record into the validator, you just give it the name of your domain. The validator fetches the record from your DNS directly and confirms that it makes sense.

You should also register your newly SPF-enabled domain at the online registry. Go to http://www.spftools.net/ where over 200,000 other domains have self-registered. By some estimates, the true number of SPF-enabled domains has already crossed the one million mark.

Keep Track of Violations

SPF-enabled receivers should now be able to read your record and test incoming mail against it. On today's Internet, we can assume that our domains are being routinely spoofed by malware. If you want to find out who's spoofing your domain, you can add a mechanism to log any spoof attempts seen by SPF-enabled receivers. You'll need a nameserver that logs all queries made against it; and you'll need to set up a domain that causes queries to hit that nameserver. Suppose that domain is `logger.sender.com.` You can add an `exists` mechanism before the terminal `all` so your SPF record looks like this:

```
v=spf1 a mx exists:%{s}.S.%{i}.I.logger.sender.com ~all
```

The macros `%{s}` and `%{i}` expand to the sender address and the client IP address, respectively.

Macros: the macro feature lets you insert data into the domain arguments for mechanisms. `%{i}` expands to the client IP address. `%{s}` expands to the full sender email address. See the Protocol Specification.

You may find that you have legitimate end-users using an outbound gateway you had not previously identified: maybe they're using some third-party SMTP server, or maybe you didn't know about a legal outbound route. Setting up a logging `exists` helps you discover these cases. In the first case, where an end-user is using an unapproved SMTP server, your job is easy: tell the end-user that for security reasons he now needs to use `sender.com`'s SMTP gateways to send `sender.com` mail. In the second case, you can just add the new outbound gateway to your SPF record.

If you don't want to set up DNS logging yourself, you can use a third-party logging service.
See `http://spf.pobox.com/certification.html`

Dear Customer,
isp.com prides itself on keeping up to date with the latest security practices. As part of our work with our peers and partners to help fight spam, we are strengthening our security policies. We appreciate your cooperation in making the following changes:

– please ensure that your SMTP server is set to `smtp.isp.com` port 587…
– please run Windows Update…

When you send mail from your `username@isp.com` address, it is important that you send it through *isp.com*'s SMTP servers. This helps authenticate your messages and ensure that they are correctly delivered. If you send mail through any other SMTP servers, antispam software on the receiving end may classify your messages as spam.

Loose Ends: Publishing Records For Hostnames

This example has demonstrated how to publish an SPF record for `sender.com` that works for both SPF Classic and Sender ID. That's an excellent start, but it doesn't end there! Your primary domain name is certainly the first thing you should take care of. But to be really thorough, you should put SPF records on the hostnames under `sender.com`.

In this example we've seen two hosts: `corp.sender.com` and `smtp.sender.com`. Both are used for outbound relaying: they send mail from addresses at `sender.com`. But they may also send non-delivery notifications (NDNs) directly. Messages from mailer-daemon are typically addressed from the hosts themselves, and look like this:

```
HELO smtp.sender.com
MAIL FROM:<>

From: <mailer-daemon@smtp.sender.com>
```

To accommodate cases like that, you should publish SPF records for `smtp.sender.com` and `corp.sender.com`. They'll be much simpler than the record for sender.com. In fact, all they have to say is this:

```
smtp.sender.com IN TXT “v=spf1 a -all”
corp.sender.com IN TXT “v=spf1 a -all”
```

That means: only `corp` and `smtp.sender.com`, respectively, are allowed to send mail from `corp` and `smtp.sender.com.`

If `corp.sender.com` sends mail that looks like this:

```
HELO corp.sender.com
MAIL FROM:<>
From: Mail Delivery Subsystem <mailer-daemon@corp.sender.com>
```

Here is how your record might look:

```
corp.sender.com TXT “v=spf1 a -all”
```

Loose Ends: Deferral Relays

If a message cannot be delivered, it is queued for later retry. All sane senders do this. But some sites practice *deferral relaying*: instead of queuing undeliverable messages on the sending server, they move messages to a different machine. This is done for performance reasons. The main sending servers don't get clogged with queued messages, and the deferral servers can focus on retrying messages at a more leisurely pace.

If `sender.com` practices deferral relaying, you should add its deferral relays to its SPF record.

Deferral Relaying: when a message is not immediately deliverable, queue it on a dedicated retry server instead of on your main sending server.

If `deferrals.sender.com` also sends mail that looks like this:

```
MAIL FROM:<somebody@sender.com>
From: Some User <somebody@sender.com>
```

Or like this:

```
MAIL FROM:<>
From: Mail Delivery Subsystem <mailer-daemon@corp.sender.com>
```

We could update the previous examples to say:

```
corp.sender.com TXT “v=spf1 a a:deferrals.sender.com -all”
smtp.sender.com TXT “v=spf1 a a:deferrals.sender.com -all”
sender.com      TXT “v=spf1 a mx a:deferrals.%{d2}   ~all”
```

To Fail or not to Fail?

If you look at other sites with SPF records, you'll find that some of them end in `?all`, some of them end in `~all`, and some end in `-all`. What should you do?

It depends. This is a tradeoff situation: you have to balance competing concerns. Conservative publishers might start with a `?all`, move through `~all` as conditions change, and (if all goes well) stabilize at `-all`. (“Conditions change” means users switch to the approved outbound SMTP relay, forwarders start prepending headers and implementing SRS, and you start signing with DomainKeys.) If you are very concerned about phishing, publish a `–all` right away and accept that there may be some false positives due to noncompliant forwarders who are slow to upgrade. Otherwise, use a `~all`.

If…	**then set**
Deliverability is mission critical…	?all
Phishing is a major concern…	–all
Your users are still using 3rd party SMTP servers…	?all
All users are known to be compliant…	–all
You are worried about non-SRS forwarders…	?all
“Enough” forwarders have adopted SRS…	–all
(or you want to encourage them to)	
The domain never sends mail…	–all
You need something in between ?all and -all…	~all

Yes, this is yucky. I'm sorry. I wish it weren't. –Meng

Expect to use DomainKeys or other crypto

Cryptographic authentication is following in the footsteps of IP-based authentication. When it becomes possible for your MTAs to sign outbound messages with DK or a similar cryptographic application, you should do so. Solutions like SES and BATV are also maturing rapidly, and offer comparable functionality but with a different cost/benefit structure. A later version of this whitepaper will describe cryptographic solutions when they mature.

gmail.com and yahoo.com have started signing outbound mail with DomainKeys. Other ISPs and email providers are poised to follow suit.

http://www.elan.net/~william/emailsecurity/emailsignatures-comparisonmatrix.htm compares a number of cryptographic signature schemes.

DEPLOYMENT: FOR EMAIL RECEIVERS

Two Sides of the Coin

Sender authentication can be put to two uses. Messages from trusted senders that pass authentication can be saved straight to end-user inboxes, bypassing expensive and potentially erroneous content filtering. Messages that fail authentication can be identified as forgeries and rejected at SMTP time, silently discarded, or filed to a junk folder.

In the case of a false positive, the argument for system integrity weighs in favour of rejection at SMTP time. Legitimate senders deserve to know when their mail isn't getting through.

Recording Trusted Senders Who Passed Authentication

If a joint test shows that the sender is trusted and the message is authentic, a receiving system should communicate that fact to end-users. A receiving MTA should record the test results in a header for consumption by a downstream MUA.

The industry hasn't yet standardized on a way to do this. Murray Kucherawy of Sendmail has drafted a specification for an `Authentication-Results:` header. I haven't heard any complaints to date.

Whitelisting Incoming Forwarders

There are lots of forwarding services out there. Perhaps the best known forwarders among the technical community are acm.org and pobox.com. Similar alumni forwarding addresses are offered by universities. Domain hosting services quietly offer email forwarding as part of their standard service set. And uncounted more ad hoc setups exist, running out of a `.forward` file set up by an end-user who may have forgotten all about it.

It is difficult to comprehensively whitelist all incoming forwarders. An ISP has no *a priori* knowledge of its end-users' third-party `.forward` files. But it can whitelist well-known forwarders *en masse* thanks to trusted-forwarder.org. That domain is both a DNSWL and an RHSWL of well-known forwarding systems. It includes acm.org, pobox.com, and many major alumni forwarding and domain hosting services. It attempts to factor out the tractable parts of the forwarding problem, and leaves only the ad hoc, unpredictable `.forward` scenarios.

Trusted-forwarder.org has been in existence for over a year. It is well maintained by Wayne Schlitt, a core member of the SPF project. All SPF implementations are strongly encouraged to use it until cryptographic solutions solve forwarding for good.

One school of thought says that because this sort of forwarding operates on behalf of the receiver, it is the duty of the receiver system to whitelist the intermediate forwarder. Another school of thought contends that it is the responsibility of forwarders to accept responsibility for reinjecting messages into the mailstream, and therefore forwarders should be prepending headers and doing SRS. Either way the situation is messy, and nobody wants the hot potato. Both schools are right to some degree. The pragmatist asks: what can we do if the other side is obstinate?

Progressive forwarders will prepend headers and do SRS voluntarily, without taking the position that receivers should whitelist them; they will also publish SPF records for their HELO domain names to make whitelisting easier.

Progressive receivers will whitelist forwarders voluntarily by IP address or by validated HELO domain name, without taking the position that forwarders should do SRS.

If the entire Internet were this progressive and polite, we'd have solved spam a long time ago.

Note that whitelisting means different things to different people. A conscientious forwarder may aggressively spam filter, and only let through good mail; in that case, it would be safe to give them a free pass. But a forwarder that doesn't filter should be subject to the same content filtering as any other untrusted sender.

What To Do About Forgeries

If the authentication test returns a FAIL result, what should you do? A receiving MTA can reject the SMTP session or use the FAIL as a part of a spam scoring scheme. If the message is not rejected at SMTP time, the receiving MTA should record the test results in an `Authentication-Results` header for consumption by a downstream MUA.

Deployment: For ISPs and Enterprises

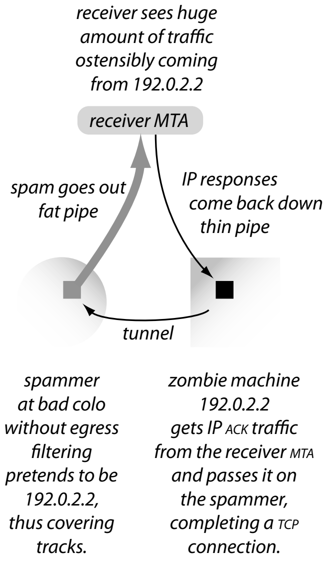
The preceding sections “for Senders” and “for Receivers” apply to ISPs and enterprises as well.

Complementary considerations for ISPs

While a detailed deployment discussion of complementary technologies is outside the scope of this document, ISPs are strongly encouraged to adopt them as part of a balanced antispam strategy.

Outbound Port 25 blocking restricts direct-to-MX spam from zombies. ISPs should block port 25 outbound on consumer-class connections by default. (Customer churn is always less than feared.) If a customer needs the port unblocked, ISPs can do so on a case-by-case basis or offer a different class of service. Business-class connections should leave port 25 unblocked by default under the assumption that customer organizations – acting as ISPs themselves – run their own MTAs and internally block port 25.

Inbound Port 25 blocking is also strongly recomended to counter asymmetric IP routing attacks.


receiver sees huge amount of traffic ostensibly coming from 192.0.2.2
receiver MTA
spam goes out fat pipe
IP responses come back down thin pipe
tunnel
spammer at bad colo without egress filtering pretends to be 192.0.2.2, thus covering tracks.
zombie machine 192.0.2.2 gets IP ACK traffic from the receiver MTA and passes it on the spammer, completing a TCP connection.

Asymmetric IP Routing Attack

Offering SMTP AUTH is essential for roaming customers who, in the stricter world of sender authentication, need to connect back to their home ISP to send mail. SMTP AUTH should be offered on port 587 because a roaming user might not be able to reach port 25. Encryption should be mandatory: the STARTTLS option is widely supported. CRAM-MD5 and port 465 are good alternatives.

Consistent Reverse DNS Naming gives receivers a way to guess if a host is a zombie or a legitimate outbound MTA. Consumer-grade machines should reside under a specific subdomain. Production MXes should never contain their IP address in the hostname and must have consistent forward and reverse DNS.

If the IP address appears in the hostname, together with a distinctive subdomain name, lots of people will recognize that as a consumer-class broadband machine that should not send mail. Conversely business-class accounts should get their choice of reverse DNS naming, and the PTR hostname should resolve back to the actual IP.

Some filters are unable to cope with embedded wildcards so the distinctive tag should appear on the right hand side.

Good:	`192-0-2-2.adsl.isp.com`
Bad:	`2-2.adsl.192-0.isp.com`

Some common subdomains:
dsl · adsl
cable · cablemodem
cust · customer
dial · dialup
dyn · dynamic
ppp · pppoe
broadband

Spam-filtering outbound mail and Rate-Limiting consumer-grade senders to a reasonable volume (perhaps 100 messages per day by default) would be an effective way to stem the flow of spam from zombies. With SMTP AUTH, you can do this based on username instead of IP address.

Detecting Outbound Return-path Forgery can be an effective optimization: if a customer node is sending mail with a wide variety of sender addresses, some of which would fail SPF tests, then it is highly likely that that node is compromised. ISPs should match return-paths to authenticated user identities (e.g. the SMTP AUTH username): this prevents cross-customer forgery and limits damage to the affected user.

The nice thing about standards is that there are so many to choose from. Furthermore, if you do not like any of them, you can just wait for next year's model.
Andrew S. Tanenbaum

Which specification?

One half of Sender ID is SPF Classic. The original SPF Classic specification was frozen in early January 2004. It has evolved in only minor details since then. It was submitted to the IETF in October 2004 and is expected to be published as an experimental RFC. When it is published, MTA vendors are encouraged to update their implementations to match it. Vendors who implement SPF Classic can indicate that they are Sender ID compliant.

See http://spf.pobox.com/rfcs.html

The other half of Sender ID is the PRA check. MTA vendors may also wish to implement that half. Specifications describing the PRA and how it fits into Sender ID were submitted to the IETF in October 2004 as well.

See http://www.microsoft.com/senderid

Conformance testing

The easiest way to ensure that an implementation is conformant, of course, is to run it through an industry-standard interoperability test suite. Certification programs for SPF and Sender ID are in the works and will be announced publicly when they are ready.

Perform SRS and prepend headers when forwarding

When automatically forwarding mail, MTAs should do two things: rewrite the return-path using SRS, and prepend a `Resent-From` header.

See http://spf.pobox.com/srs.html and http://www.libsrs2.org/

Add ESMTP support for Submitter

In server mode, advertise and accept the SUBMITTER parameter. In client mode, if the server supports it, send it.

Record authentication and policy results in the headers

It is the job of the MUA to signal trustworthiness to the end-user. It is the job of the MTA to help them do this. MTAs should record authentication results in as much detail as possible into headers. MTAs should, of course, be sensitive to header spoofing. They can do this by renaming or removing preexisting headers. See the `sender-auth-header` internet-draft by M. Kucherawy for more details. It describes the proposed header, "`Authentication-Results`".

Join the developers mailing list

SPF and Sender ID developers are strongly encouraged to send mail to `subscribe-spf-devel@v2.listbox.com.`

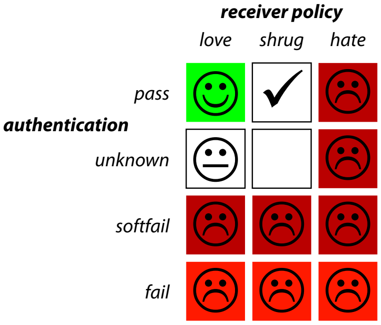
See also http://spf.pobox.com/developers-guide.html

Deployment: For MUA vendors

Displaying Authentication-Results

MUAs need the email equivalent of the HTTPS padlock icon. An MUA can rely on an upstream MTA to produce an `Authentication-Results` header. It can also perform authentication checks directly: if the message was cryptographically signed this is easy. If the message was not signed, and the MUA needs to use Sender ID techniques, this is a little more difficult.

MUAs should visually distinguish messages that are considered trustworthy from messages that lack an authentication status or that failed authentication. Note that a message should be considered trustworthy only if it passed both tests: authentication and local receiver policy. MUAs can contribute to the "local receiver policy" decision: if a message was authenticated, and the sender appears in the end-user's addressbook, that might add up to a dual-PASS. MUAs should add a clearly visible warning to messages that fail authentication and could refrain from displaying inline images by default.



		receiver policy		
		love	shrug	hate
authentication	pass	☺	✓	☹
	unknown	😐		☹
	softfail	☹	☹	☹
	fail	☹	☹	☹

Some possibilities for MUA displays.

MUAs should also consider adding a "Not Junk" folder and automatically file trusted messages into that folder. The obvious idea is to give good messages attention priority. The subversive idea is for end-users to one day say "hey, I don't even look at my Junk Folder any more, so I'll just set that to auto-delete (or reject); and now that I think about it, that's true for the regular inbox too!" The "Not Junk" folder will be all that remains.

Automatic switching to port 587

The ISP industry is moving toward the practice of blocking outbound port 25 from consumer-grade dialup and broadband nodes.

This means that roaming users will increasingly find themselves in need of an alternative port for message submission. That port is port 587, as defined in RFC2476.

At present, users need to hunt down the configuration dialog which lets them change their submission port from 25 to 587. This can be challenging: MUAs sometimes hide that configuration point in esoteric places.

MUAs should automatically probe port 587 when submitting mail. Since they already possess the end-user's username and password for message retrieval, they should have no problem performing authenticated submission with SMTP AUTH. If the 587 attempt fails they can fall back to port 25. All of this can occur behind the scenes without user intervention – which is the way it should be.

Deployment: For ESPs

When a big brand wants to mail many, many customers, it often contracts the job to an Email Service Provider (ESP). ESPs handle the mail merge, bounce management, and unsubscription functions for the brand.

For the most current version of this page, see
`http://spf.pobox.com/esps.html`

Don’t look like a phisher!

In the last few years, the chaotic world of volume-mailing outsourcing has shot itself in the foot: the standard configuration for an outsourced campaign now looks practically indistinguishable from a phishing attack. ESPs need to make their relationship with their clients a little more obvious.

Delegation

DNS comes with a delegation feature. Use it! Suppose your client is `bigbox.com`, and you’re `esp.com`. You should try to get the client to set up `esp.bigbox.com` which delegates to you; you can then use `esp.bigbox.com` in your mailings with full control of the DNS.

In the example below, bigbox.com’s DNS admins have delegated `esp.bigbox.com` to `esp.com`, so `esp.com`’s nameservers are the ones that actually answer authoritatively for `esp.bigbox.com`. This makes it easier for `esp.com` to fiddle with `esp.bigbox.com`’s SPF records and whatever else. The alternative to delegation is the infrequent and troublesome need for ESP to call BigBox once every eighteen months or so when the DNS details change.

If your client is too small or too unsophisticated to set up delegation, and if they just want you to use their name directly in their mailings, you’ll need to get them to “`include:`” you in their SPF record.

Publish Appropriately

ESPs are the one sector who might need to distinguish PRA and MAIL FROM scopes. We’ll walk through an example.

Suppose an ESP, esp.com, has been contracted by a client, `bigbox.com`. The ESP sends mail using through a server, `outmta.esp.com`. Bounces and unsubscribes are directed to `bounces.esp.com`.

The author recommends that ESPs set up their mailings to look like this:

```
MAIL FROM:<handler-23451@bounces.esp.com>
From: Big Box Stores <marketing@bigbox.com>
Sender: <mailings-for-bigbox@esp.bigbox.com>
Reply-To: <unsubscribe-23451@bounces.esp.com>
```

The PRA algorithm selects the `Sender` header.

If you wanted to distinguish PRA from MAIL FROM scope, you might use the following records:

```
1  bounces.esp.com
2    “v=spf1 a:outmta.esp.com ?all”
3    “spf2.0/pra -all”
```

Line 2 is for the MAIL FROM. The default is “?all” because for this mailing campaign and this domain, `bounces.esp.com`, we’re more concerned about mail getting through than about preventing forgeries. Line 3 is for the PRA. It states: we’ll never use the domain `bounces.esp.com` in a `From:` or `Sender:` header, only in the envelope.

```
4  bigbox.com
5    “v=spf1 a:corp.bigbox.com -all”
```

Line 4 is for the client, `bigbox.com`, which outsources all of its marketing mailings and only uses bigbox.com for corporate accounts. it is more concerned about phishing than false positives due to forwarding, so it sets `-all`.

```
6  esp.bigbox.com
7    “v=spf1 -all”
8    “spf2.0/pra a:outmta.esp.com”
```

Lines 7 and 8 are for the outsourced relationship domain `esp.bigbox.com`. it indicates that ESP is a contractor for `bigbox.com`. `esp.bigbox.com` is never used in the return-path so the `v=spf1` record is `-all`. But it is used in the `Sender` header, so the `spf2.0/pra` record overrides it to allow `outmta.esp.com`. The PRA record does not end in a `-all` because, again, for this mailing we’re more concerned about mail getting through than about forgery prevention.

ESPs who feel in need of further guidance are welcome to contact mengwong@pobox.com directly.

Deployment: For Spammers

Spammers are some of the most responsive players in the industry. They have been enthusiastic early adopters of authentication technologies. The "Devil's Advocate" deployment advice in this section forecasts the next few moves in the game and predicts the final steady-state scenario. It is meant to show how spammers will adapt and how the good guys can respond.

Two Types of Spammers

There are two major types of spammers in operation today. (Legitimate ESPs are not considered spammers.)

Organized crime is largely responsible for the phishing and identity-theft fraud scams we see today. Many of these criminal gangs work across international boundaries.

Commercial spammers are responsible for the less obviously criminal junk mail – they actually try to sell a product. The product may be held up in shipping, or it may not actually deliver the advertised results, but some kind of sale is usually promised.

Publish SPF and sign with DomainKeys.

In the relatively near future, a sea change will occur so that mail from senders who are not accountable will be widely relegated to third-class status.

Some receivers may look down on messages from senders who do not have SPF or DomainKeys. To be on the safe side, spammers should publish and sign.

Note to civil libertarians: you can be accountable and anonymous at the same time. We're interested in getting identities on the Internet to persist long enough to attract a stable reputation. We're not quite as interested in tying online identities to real-world identities, because support for whistleblowing is an important social value. But we want to give receivers the ability, when faced with a range of messages from senders they don't know, to apply an ordering to those messages so that senders who have taken steps to distinguish themselves from spammers get some kind of priority over senders who have not taken those steps. A sender might voluntarily choose to tie their online identity to their real-world identity in the hopes of improving their deliverability; however, that decision should be entirely up to senders, and there should be ways for senders to distinguish themselves from spammers without requiring real-world identification.

Stop forging random domains.

Sure, you could forge domains that don't have SPF records, but after a while receivers will adapt by simply third-classing mail that doesn't authenticate. So this is only a temporary measure.

Buy your own domains.

There are two main strategies. You could buy a domain, bring it online, and immediately spam with it. This works because reputation service won't know anything about those new domains, so if the receiver has a default-accept orientation, the mail might still get through. As this strategy evolves, receivers might learn not to accept mail from brand new domains. This leads to the second strategy: buy a domain, leave it mostly dormant for some time, and then spam with it in hopes that it'll now look more reputable than a brand-new domain. Because most new spam domains are bought with stolen credit cards, this strategy may be more costly unless

According to some reports, over a thousand domains are registered each week and used to spam.

you can find a sloppy DNS registrar who'll leave up domains that were registered with stolen cards. But then, using such a registrar might negatively affect your reputation. The author recommends that you experiment and see how it works out.

Reuse an expired domain.

You can effectively hijack someone else's good reputation by finding a respected domain that is recently expired, and send mail using that name. But reputation systems are likely to pay attention to domain expirations and transfers, and null out the reputation on a domain that has changed hands.

Try to buy accreditation.

Accreditation schemes may become widespread. The essential ruse is to look like a good guy whose domain is simply new to the Internet, so you can try to sign up with accreditation services that don't do very good due diligence. But accreditation services that don't do very good due diligence are likely to attract a poor reputation themselves.

Try to fake out reputation services.

Reputation services that depend on end-users are subject to attack, because you can pretend to be an end-user, and vote your spam as ham. In fact, you can pretend to be several thousand end-users. But as reputation systems grow more sophisticated, voters will be themselves subject to reputation, so an attack may not succeed unless you can completely overwhelm a reputation service's population.

Zombies should spam only their friends and family.

Botnets of infected zombies are your major asset today. In the future, you will need to upgrade your zombies to figure out SMTP AUTH credentials, label the spam as being from the actual zombie user, route it through the ISP's MTA, and only direct it to people in the end-user's addressbook and mailbox. Spam sent through other routes and to unrecognized recipients will be unlikely to be delivered or read. Good ISPs will counter with rate-limiting and outbound filtering.

Spread FUD about the edge cases.

FUD: (n) Fear, Uncertainty, and Doubt.

None of the approaches are perfect. A message could be forwarded through a site that does not perform SRS and does not prepend `Resent` headers; that message could then pass through an MTA that munges the content for perfectly good reasons. This corner case is a favourite of technical perfectionists who use it to argue that one can never reliably reject a message based on sender authentication. If this point of view gains widespread public acceptance, you will be able to continue to spoof messages.

Deployment Roadmap

Publishing SPF: At the MAAWG First General meeting on November 2 2004, a majority of members voluntarily agreed to publish SPF records by the end of December 2004. It is up to each member to decide which default to use. Most conservative members may wish to use `?all` (neutral) at first; that's fine. More aggressive members may wish to use `~all` (softfail). See the tradeoffs on p 15.

Support for TXT: Managed DNS providers should support TXT entries by the end of Q1 2005. Providers should link to a well-known third party wizard instead of offering their own.

Crypto Signing: Some members are planning to sign messages with DomainKeys on an experimental basis in 2005 as their MTA software develops this capability.

Checking SPF and DomainKeys: A testing and evaluation program is underway. MAAWG members expect to start SPF testing incoming mail in Q1 2005 and to incorporate the result into spam scoring decisions in Q2 2005. DomainKeys checking should occur close to that timeframe also.

Forwarder SRS: Forwarders should perform SRS on forwarded mail and prepend headers by the end of Q2 2005.

Honoring Authentication Failures: If a sender domain uses a `-all` default, and if a receiver domain obtains a FAIL result, that receiver may reject the message during the SMTP transaction. It should not generate a bounce message. In Q3 2005, senders should set `-all` defaults, and receivers should move to begin to honour `-all` by rejecting. Senders who are particularly concerned about noncompliant users or forwarding false positives can define a `~all` default.

Displaying confidence to end-users: ISPs should start recording `Authentication-Results` by the end of Q2 2005. MUAs sold after that date should display a confidence mark distinguishing good senders from bad.

Requiring Authentication Passes: Spammers are expected to start registering their own domains and churning through them. The weak way to answer this is to start keeping lists of domain names to block, but that is a reactive mode of operation akin to today's IP-based blacklists. The strong answer is to only accept mail that passes authentication and reputation/accreditation tests. ISPs should • start offering a default-reject type of mailbox in Q4 2005 or sooner, • make that the default offering for new signups, and • give existing users the option to convert to default-reject.

Q4 2004	Senders and ISPs publish SPF records.
Q1 2005	MAAWG SPF testing continues. DNS hosters support TXT records.
Q2 2005	Forwarders do SRS and prepend `Resent-From`. MUAs display confidence to end-users with foldering to first-class and business-class. ISPs offer 587 SMTP AUTH for roaming users, tell new subscribers to configure that way by default. Cryptographic solutions mature.
Q3 2005	SPF results widely used in spam scoring. Senders start signing outbound mail and transition SPF records to `~all` or `-all.` Receivers check inbound signatures and honour FAILS by rejecting.
Q4 2005	ISPs offer default-reject mailboxes. Spam ends. Bill Gates gets credit.

Note: the above deployment dates are the author's recommendations. They are still incomplete, subject to further development and discussion, and have not been ratified by any industry standards body. However, if you wait for industry standards bodies to ratify things, you may be waiting a really long time. That said, you should consider joining MAAWG, which is a forum for collaboration and standards deployment in the messaging space. (http://www.maawg.org)

To stay current on these issues, join the deployment mailing list by sending mail to `subscribe-spf-deployment@v2.listbox.com.`

Contact the author for details on the Karma Project, which aggregates and multiplexes reputation feeds for convenient lookup.

See also the MAAWG Code of Conduct document and the ASTA Technical Recommendations document.

Code of Conduct document currently being revised.

ASTA doc available at http://docs.yahoo.com/docs/pr/pdf/asta_soi.pdf

1. Noncompliance with RFC2821 and RFC2822 constitutes sufficient cause to reject a message.

2. When an outbound edge MTA sends a domain name in the HELO argument, that hostname must be a Fully Qualified Domain Name (FQDN) that resolves to the IP address of the MTA.

3.1. Every outbound edge MTA must have a reverse DNS (PTR) record that resolves to a hostname that in turn resolves back to the IP address of that MTA. To avoid looking like a consumer-grade machine, it should not contain more than one octet of its IP address in its hostname.

3.2 Conversely, dynamically assigned consumer-grade IPS should obviously contain two or more octets of their IP addresses in their PTR hostname. See p 17.

IP addresses may be encoded in decimal or in hex. For example, `192-0-2-2.adsl.example.com` or `c0000202.adsl.example.com`

4.1. A dynamically assigned consumer-grade dialup or broadband node should not expect to be able to send mail directly to unrelated receiver MTAs over port 25. ISPs should block outbound port 25 either at the network routers or inside the customer broadband modem. ISPs should offer a message submission server for end-users to use as an outbound mail relay.

4.2. As a corollary for home users: unsecured wireless routers can be configured to block port 25 too. Meng personally submits mail over 587, so blocking port 25 in his little Linksys gizmo (A) doesn't disrupt anything, and (B) makes him feel better about leaving it unsecured and open to the public.

5. If an SMTP client appears to be a consumer-grade dialup or broadband node, and if that SMTP client does not demonstrate accountability, receivers may reject the connection.

Clients demonstrate accountability by publishing SPF records or signing with DomainKeys.

Spam is not a problem to be solved.
It is a phase to be outgrown.