

SPF, MTA's and SRS

Meng Weng Wong
(Reprinted from Linux Journal)

Sender Policy Framework (SPF) takes aim at the practice of return-path spoofing, a technique employed by worms, viruses and other senders of unwanted mail. SPF consists of two parts. First, domain administrators publish SPF records in the DNS. Those records describe the servers the domain uses for outbound mail. Then they are read by SPF-enabled MTAs. Mail coming from a server not described in SPF can be considered forged.

This article, the second of a two-part series, explains how to add SPF capabilities to your mail server. It also discusses how e-mail forwarding and Web-generated e-mail services can adjust to SPF by performing sender rewriting.

This article was written in early February 2004 and reflects the state of the Internet at that time. The MyDoom worm, a virus that spoofs return-path addresses, recently had littered millions of mailboxes with bogus bounce messages.

It's Your Turn

Last month I described how to construct an SPF record, and DNS administrators all over the world responded. First, they published records, then crossed their fingers and waited. What are they waiting for? They're waiting for you. They've made it possible for you to distinguish their legitimate mail easily from forgeries. Now it's your turn to help them cut down on bogus bounces and abuse reports. If you're just tuning in, see the on-line Resources section for last month's article and an easy Web-based SPF wizard.

Adding SPF to Your MTA

The major mail transfer agents (MTAs) in the Linux world are Sendmail, Postfix, Qmail and Exim. Although most antispam vendors already have SPF support included in their products or plan to add it in their next release, MTAs tend to want to leave that task up to you. Most MTAs offer an interface into which you can plug your antispam tool.

SPF can be made to work in your MTA in two ways. If you're the kind of sysadmin who prefers to compile your own software, start at the SPF downloads page. There you can find the SPF plugin that's right for your MTA, plus detailed installation instructions. If you prefer to manage your software using a package system, you may find an SPF-enabled version of your MTA already built and ready to install.

Most of the plugins rely on the reference Perl library Mail::SPF::Query. You can install that library directly from CPAN, or you can try to find a package for it. It provides a simple program to run SPF queries at the command line. It also provides a simple dæmon that handles SPF query requests over a UNIX domain or inet socket.

By default, most of the plugins tell the MTA to reject messages that fail SPF tests and add a Received-SPF header to the rest. Conservative installations may prefer to add the line Received-SPF: fail instead of rejecting. This configuration option is described in the plugin documentation.

Sendmail

Sendmail's plugin interface is called Milter (see on-line Resources). Recent Sendmail versions have Milter capability compiled in by default. Sendmail talks to Milter through a socket interface. Sendmail tells Milter about the incoming SMTP transaction, and Milter tells Sendmail what to do. Milter runs as a dæmon and needs to be started separately.

SPF, MTA's and SRS

Meng Weng Wong
(Reprinted from Linux Journal)

Two Milters should be available at the SPF Web site: one in Perl and one in C. The Perl version is a little more mature, but if you need speed, the C version may be a better choice.

To make Milter work with Sendmail, add a couple of lines to your sendmail.mc file, rebuild sendmail.cf and restart Sendmail.

If you'd rather not use Milter, libspf comes with a patch that integrates SPF directly into Sendmail.

Postfix

Postfix 2.1 comes with a policy daemon interface. It works much like Milter does: Postfix connects to the daemon and provides a play-by-play commentary, and the daemon returns an action to Postfix. If you're running a recent development snapshot of version 2.0, make sure you're using 2.0.18-20040122 or later.

Policy daemons are configured in main.cf and master.cf. They are managed by Postfix, which starts and stops them as needed, so you don't need to worry about that. The Postfix policy daemon is written in Perl and calls the standard Mail::SPF::Query library.

Exim

Exim 4 introduced Access Control Lists (ACLs), a powerful and compact mini-language for making antispam and other local policy decisions. The ACL code that handles SPF for Exim is only about 12 lines long.

You need to install the Mail::SPF::Query library and run its SPF daemon, which listens on a socket. The SPF ACL connects to the spfd and reads it the client IP, HELO argument and MAIL FROM sender address. It then receives an SPF result, a response for the SMTP server and a Received-SPF header line. You need to start the spfd separately.

Qmail

Qmail does not have the same kind of plugin interface that the other MTAs do. Instead, SPF provides a patch that integrates SPF directly into Qmail. In addition, many Qmail users screen their mail with qpsmtpd: if you do, SPF is a plugin you can turn on easily.

James Couzens is the primary author of the C SPF library. libspf comes with a patch for Qmail and for other MTAs as well.

Testing the Plugin

Once you've installed the plugin and turned it on, you should perform two tests. First and most important, legitimate mail needs to get through. If something broke, maybe you're not running something you need to—double-check. If it's still broken, back out the patch and report your experience to the spf-help mailing list.

Second, confirm that forged mail is rejected. If you can speak SMTP by hand, engineer a message with MAIL FROM:<linuxjournal-test@altavista.com>. The domain altavista.com is not used for mail, so it always returns a FAIL message. They have asked that test messages contain the word test. This can be tricky to execute because if they recognize a trusted client, both your MTA and SPF will turn a fail into a pass. Therefore, don't telnet to localhost; use your machine's actual hostname, and if possible try to open the connection from an outside host. If you receive a 550 response and an error message that refers to spf.pobox.com/why.html, it's working.

SPF, MTA's and SRS
Meng Weng Wong
(Reprinted from Linux Journal)

If you use a secondary MX, tell your SPF client not to reject its mail. How to do this is described in detail in the installation instructions for your plugin.

Received-SPF: What the Codes Mean

You should notice that your mail now contains a Received-SPF header that carries a number of result codes:

- **NONE:** the domain does not publish SPF records. Your MTA should proceed as usual.
- **PASS:** the mail is not forged, but that doesn't mean it's legitimate. Remember, spammers can publish SPF too. You still should test its domain against a right-hand-side block list (RHSBL). But if the sender is on your trusted whitelist, you can skip further antispam checks with confidence.
- **FAIL:** the mail is a forgery, and you can reject it with confidence. There is a miniscule chance the message is legitimate but was sent by a misconfigured sender. In that case, the error message they receive tells them they need to configure their MUA with SMTP AUTH. SPF's design philosophy is that it's better to fail obviously with a hearty error message than to risk silently burying mail in a spambox.
- **SOFTFAIL:** the message could be a forgery, but the domain's ISP is working on switching its users to SMTP AUTH, so the message could be legitimate. You should accept the message, but subject it to more stringent antispam checks.
- **NEUTRAL:** the domain just has started down the road to SPF, and their default response is ?all. They would like you to pretend the response was NONE while they consider moving the default toward SOFTFAIL and FAIL. Big ISPs with millions of users move slowly; it's not their fault.
- **ERROR:** there was a temporary DNS lookup error. Normally, your MTA should return a 450 temporary failure when this happens.
- **UNKNOWN:** a permanent error caused the SPF lookup to abort; perhaps there was a syntax error in the record, or maybe the record pointed to another domain that doesn't have an SPF record.

The Price of SPF

In the past ten years we have grown tremendously dependent on e-mail; we are made aware of just how dependent we are every time a worm hits. Analysts routinely announce that spam and viruses cost the economy billions of dollars. The success of SPF shows that people are desperate for change.

But, change has its own price. If there were such a thing as a painless solution to spam, we already would have adopted it. The war on spam has dragged on so long in part because the best experts on spam simply could not agree on exactly what trade-offs they wanted to make, but that phase of debate is drawing to a close. In every antispam future they have discussed, sender authentication is the first and fundamental step. Now, many possible sender authentication models are available, but the designated-sender scheme that SPF provides is probably the easiest to implement.

Cryptography definitely is in our future, but it's not here yet. Like first aid, SPF offers immediate benefit, and it's something we can do right away.

SPF, MTA's and SRS

Meng Weng Wong
(Reprinted from Linux Journal)

What is the price of SPF? Every designated sender scheme breaks two things. First, SPF breaks verbatim e-mail forwarding (Figure 1). Services that provide permanent e-mail addresses, such as pobox.com, are used to forward mail the way UNIX .forward and /etc/aliases files do. When the mail leaves their servers, the return-path address in the envelope is unchanged. But in an SPF world, resent messages now look a lot like forgeries. To fix this, forwarding services need to rewrite their return paths. So do other sites that depend on .forward and /etc/aliases to send mail off-site.

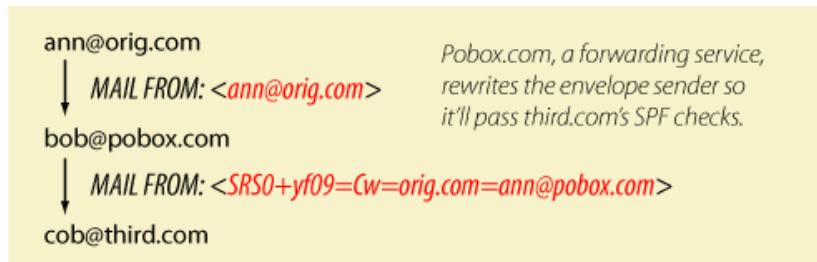
Verbatim email forwarding and Sender Rewriting Scheme



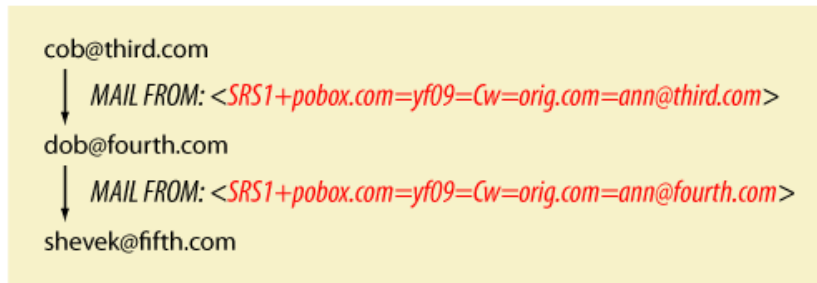
In traditional or verbatim forwarding, the return-path is preserved in the outgoing message. In an SPF world, forwarders need to rewrite the return-path to stay in good standing. The perl library *Mail::SRS* and the C version *libsrs* help perform the needed transformation. Full-time forwarding services may prefer to integrate this logic directly into their MTAs. Smaller sites don't have to: they can just call the *srs* utility which does all the work.

```
bob@pobox.com's .forward file:  
"/usr/bin/srs cob@third.com"  
  
pobox.com's /etc/aliases file:  
srs0: "/usr/bin/srs -reverse"  
srs1: "/usr/bin/srs -reverse"
```

For more information, see
<http://spf.pobox.com/srs.html>



The rewritten return-path contains the original sender so whitelisting can still work, albeit with some tweaking. To prevent bad guys from using forwarders as an open relay, SRS adds a hash (yf09). It also adds a timestamp (Cw) which causes addresses to expire. If cob@third.com is undeliverable, the bounce goes back to pobox.com which forwards it back to orig.com. Note that no escaping is needed.



What if there are multiple forwarders? No problem: the SRS0 marker tells other forwarders that SRS has already happened. The second forwarder changes SRS0 to SRS1, slaps on the first forwarder's domain, and leaves everything else untouched. Third and subsequent forwarders just change the domain at the end. This keeps address growth in check. Mail::SRS also provides a database-backed version that guarantees short addresses.

Figure 1
Old-school e-mail forwarding breaks under SPF.

The solution is called SRS, sender rewriting scheme. It encapsulates the original sender address in the rewritten, SPF-compliant, return address. If a message should bounce, it comes back to you, and you unwrap the address and forward the bounce back to the sender. Forwarding services would have to do this even in a world without SPF, because ISPs already are performing pseudo-SPF checks. SPF simply gave everyone a standard way to do what they already were doing piecemeal. In the

SPF, MTA's and SRS

Meng Weng Wong
(Reprinted from Linux Journal)

same way that responsible sites closed down their open relays over the past few years, in the coming months responsible sites will begin to operate SRS-compliant forwarding; pobox.com already is doing SRS, and other forwarding services are expected to follow.

The good news is the community that developed SPF already has produced SRS code for your MTA. Those patches are available from the same place you got your SPF patches. By the time you read this, they even might be bundled into your MTA. The goal is for the average installation to be able to upgrade to the latest version and have SRS magically work (see Resources).

So, this solves the e-mail forwarding problem. Getting SRS into the field is simply a matter of time. But SPF also breaks Web-generated e-mail. Greeting card sites and "e-mail me this news article" sites tend to use your e-mail address not only in the From: header but in the envelope sender too. In SPF terms, that kind of behaviour is indistinguishable from forgery.

To solve this problem, those sites can do one of two things. First, if the mail they send isn't that important, they can set the return-path address to nobody@example.com and eat the bounces. Newer, more progressive sites, such as Orkut, already do something like this. But if the mail is important, was sent on behalf of a user who was logged in to the Web site properly, and if the Web site had previously confirmed the user's e-mail address, then the Web site could perform SRS on itself—encapsulating the user's return address so that bounces would be properly forwarded.

What about the transition period, you ask. Won't there be a time of disruption while the forwarders groan their way toward SRS-compliance? What about the sites that are unwilling or slow to adapt?

Well, here's a little secret. We have a fairly good idea who the major culprits are; we know, for instance, that eBay sometimes sends mail with a legitimately forged envelope return-path. The people who developed SPF use eBay, too, and they don't want to lose e-mail any more than you do. So they came up with a hack. They set up a whitelist that identifies all these legitimate forgers; pobox.com is on the list, as are acm.org, eBay and the newspaper Web sites that do "e-mail me this article".

Every SPF client we've talked about in this article knows about that whitelist. Every SPF client we know of gives that whitelist a chance to override a fail. If your mother sends mail from her AOL account to your acm.org address, your SPF client accepts that message, even though it's technically a forgery. (If you get forwarded mail through a system that's not on the list—from, say, a friend's home Linux box—you should whitelist that box in your MTA.) When acm.org implements SRS, the problem will go away.

SPF's critics tend to say "it breaks forwarding". The SPF community rose to the occasion and did their best to ease the transition. They offered two solutions, one short-term and one long-term, that meet in the middle. Together they sugarcoat the bitter pill.

Change means pain. The transition to an SPF world won't be painless, but it's like the pain of an injection that makes the illness go away. E-mail is very sick. Some say it will not survive spam, but I don't agree. I think SPF will set it firmly on the road to recovery.