Taughannock Networks

Selective Sender

A Taughannock Networks Mini-Paper
January 2004

Several proposals for Lightweight MTA Authentication Protocol (LMAP) have been gathering attention of late. They all define ways for a domain to specify that some particular IP addresses are allowed to send mail for that domain, and others aren't.

LMAP has a variety of technical problems because there are surprisingly many ways that mail can be sent from unexpected places.

Selective Sender, a simpler scheme that has been proposed before under other names, is much simpler, It just declares that some IP addresses should be sending mail and some shouldn't, without stating any opinions about what domains should be in the return addresses of that mail. SS doesn't try to stop spoofing, one legitimate sender pretending to be another, but it does let network managers tell the world where the real mail servers are so that mail from other parts of the network can be rejected or at least treated with greater scepticism. (Mail systems with roaming users may well want to accept mail from anywhere so long as the sending host proves its *bona fides* via SMTP AUTH or presenting a satisfactory TLS certificate or some other validation scheme.)

Since SS is keyed to IP addresses, its model is rDNS. I'll just use IPv4 rDNS here, but the extension to IPv6 should be straightforward.

First try:

Add text records to rDNS, like this to say that 10.1.2.3 sends mail and 10.1.2.6 doesn't:

```
3.2.1.10.in-addr.arpa IN TXT "SS: YES"
6.2.1.10.in-addr.arpa IN TXT "SS: NO"
```

That's about as simple as it can get, and it makes lookups quite easy, but it has the disadvantage that in the common case that most of the range should return NO and only a few adresses YES, it would be nice if there were some way to say that all of 10.1.2/24 is NO except for a few addresses marked specially. Unfortunately the DNS doesn't make that easy.

One approach would be to make missing SS data mean NO. The problem is that provides no way to tell an IP address marked NO from a range for which the owner hasn't provided data.

Another approach would be to use DNS wildcards:

```
*.2.1.10.in-addr.arpa IN TXT "SS: NO"
```

Taughannock Networks provides strategic consulting, design, and prototype implementation on e-mail and Internet infrastructure. +1 607 330 5711 or info@taugh.com

The peculiar definition of DNS wildcards keeps this from working as well, because any address that has a PTR record (which should be most of them, one would hope) won't return a wildcard, since wildcards apply only to names with no data at all.

Another possibility would be to move SS data to its own part of the in-addr zone, or to make SS ranges for each /24, like one of these:

```
3.2.1.10._ss.in-addr.arpa IN TXT "SS: YES"
*.2.1.10._ss.in-addr.arpa IN TXT "SS: NO"
```

```
3._ss.2.1.10.in-addr.arpa IN TXT "SS: YES"
*._ss.2.1.10.in-addr.arpa IN TXT "SS: NO"
```

The first has the disadvantage that the entire rDNS delegation tree has to be duplicated, which would be an administrative nightmare, while the second handles wildcards adequately, at least at the /24 level, but is just plain ugly.

So my SS proposal is the simplest, where each address has its own SS record or possibly, the /24 version just above.