INTERNET-DRAFT
Category: Experimental
Expires: October 22, 2004

John R. Levine
Taughannock Networks

April 22, 2004

A Flexible Method to Validate SMTP Senders in DNS
draft-levine-fsv-01.txt

Status of this Memo

This document is an Internet-Draft and is subject to all provisions of Section 10 of RFC2026.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet- Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/1id-abstracts.html

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html

Abstract

Flexible Sender Validation (FSV) is a lightweight validation technique to detect and deter some kinds of e-mail address forgery. It publishes information in the DNS about IP addresses authorized to send mail for a domain, one in a family of IP based mail validation proposals dating back to Paul Vixie's original in 2002[5]. FSV uses redundant copies of IP data to permit both efficient use by very high-volume mail servers, and simple implementation on low to moderate volume mail servers.

1. General Considerations

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

in RFC 2119 [1].

This document was written in connection with the Lightweight

Mail Authentication Protocol subgroup of the IRTF Anti-Spam Research Group.

2. Address Forgery

Although SMTP has never provided any security against forged return addresses in e-mail, only in recent years has forgery become a significant problem in Internet e-mail. Forgery includes ``joe jobs'', forged mail sent to discredit the nominal sender, ``phishing'', mail impersonating an organization with whom the recipient does business, and random spam, viruses, and worms, to hide the true location of the sender. Many organizations mandate by policy that all outgoing mail is sent through the organization's own mail servers, in which case forgery can often be detected by checking whether a message was sent from an authorized IP address for the domain.

Even when all of a domain's mail comes from known specific addresses, there are situations when their mail can legitimately be sent from elsewhere, e.g., mail forwarders. This document does not address forwarding and related issues such as roaming users, electronic greeting cards, and courtesy web page notifications, because they've been beaten to death elsewhere and the workarounds for FSV are the same as for other IP validation systems such as DMP, RMX. and SPF.

3. DNS considerations

FSV works by adding records to a domain's DNS zone that describe the IP addresses that send the domain's mail. Previous sender validation schemes have used two styles of DNS records, a ``block'' record that contains all of the IP addresses for a domain, or ``factored'' records where there is (conceptually at least) a separate record for each IP address and every domain.

Block records have the advantage that a DNS client can fetch all of the data for a domain in a single request. They have the disadvantage that they need some sort of internal structure to represent lists of IP address ranges which the DNS client has to parse, and that the record for a domain with many servers might exceed 512 bytes, requiring a TCP rather than UDP transaction.

Factored records have the advantages that they are easy to test using logic similar to that used to check DNSBLs (DNS blacklists, such as the MAPS RBL), and each individual record

is small, so it will always fit in a UDP packet. The have the disadvantage that large domains can potentially require vast numbers of them, and that it can be hard to tell whether the absence of a factored record for a particular IP means that the IP isn't authorized for the domain, or that the domain hasn't published any records at all.

Rather than trying to resolve the tension between block and factored records in one direction or the other, FSV panders to all factions by providing both.

Since block records can be large enough to require a TCP transaction, clients SHOULD use block records if the data from the record will be cached within the client for a substantial amount of time (up to the TTL of the record.) Clients that don't cache SHOULD use factored records since they will normally be cached within local DNS caches.

4. Mail server considerations

Depending on the design of a mail server, either block or factored records may be handled more efficiently. High-volume mail servers often run many threads within a single process, with each thread managing an SMTP session. Those servers would prefer block records since a domain's data would be parsed once and then saved in the server and reused by any subsequent SMTP session in the same server without needing any more DNS traffic.

Low to medium volume mail servers typically start up a separate copy of the SMTP daemon for each SMTP session. Since they will usually only look up one (domain,IP) pair per session, they work better with factored records which avoid the fixed overhead of retrieving and parsing block records only to throw the parsed data away after a single use.

5. Identifying the relevant domain

For efficient implementation, the domain on which FSV keys is the domain in the envelope return address given in the MAIL FROM command in the SMTP session. For bounce messages with a null return path, the host name in the HELO or EHLO command may be used instead.

If HELO/EHLO arguments are to be validated, FSV data must be present for all host names used in HELO/EHLO commands as well as for domains in mail. Since a server typically has only a single IP address, its FSV data will be small. Or, in the common case where the hosts are within the domain used in mail, the hosts can use the domain's FSV data via a CNAME record.

6. Structure and usage of FSV DNS records

FSV records are stored in a pseudo-subdomain called _fsv. The FSV records for example.com would be in _fsv.example.com.

The block records are stored under the _fsv name itself as a TXT record and an A record. The TXT record contains a series of character strings, each of which is either an IP address or a CIDR range. An IP address is either an IPv4 address

consisting of four digit strings separated by dots, or an IPv6 address consisting of eight hex strings separated by colons. A CIDR range is an address followed by a slash and a number representing the size of the range. No other characters such as spaces, comments, or other punctuation are permitted in the FSV TXT record. A TXT record with impermissible characters or other format errors such as more or less than four components in an IPv4 address, more or less than eight components in an IPv6 address, a non-digit or a component greater than 255 in an IPv4 address, or a non-hex digit or a component greater than FFFF in an IPv6 address, is discarded. The two low octets of the A record at _fsv contains the number of text strings in the corresponding TXT record so that if, for example, the TXT record contains six strings the A record contains 0.0.0.6. The high two octets are reserved for future use and must be zero.

The reason for the redundant A record is twofold. One is as an extra check for possible truncation of a potentially large TXT record. The other is as an easily and efficiently testable indication that FSV data is present for a domain. If a domain exists but sends no mail, its _fsv TXT record contains a single null string and the A record contains 0.0.0.0.

Other than standard CNAME records, FSV provides no facilities for nesting, indirection, range merging, or any other operations on its contents. A CNAME record can be used if a domain's FSV data is identical to another domain's data. Otherwise, each domain must list its full set of servers. If a domain's data is derived from another domain's data or its own MX data, the domain's management is free to use any software it wants to construct the FSV data, but that process is invisible to and of no interest to FSV clients. FSV TXT data is deliberately in a trivial format that can easily be parsed by an obvious state machine in a single pass without backing up. This makes it straightforward to diagnose and reject invalid data.

Factored records are represented as subdomains under _fsv using the standard encoding from rDNS and DNSBLs. IPv4 addresses are reversed by component and prepended to the _fsv name, so that for example, the factored record to check whether address 10.11.12.13 is valid for the domain example.com is 13.12.11.10._fsv.example.com. IPv6 addresses are encoded similarly to IPv6 rDNS, by reversing the hex digits and appending _ip6, so that the record to check whether address 4321:0:1:2:3:4:567:89ab is valid for example.com would

be (this name is broken into two lines for typographical purposes):

b.a.9.8.7.6.5.0.4.0.0.0.3.0.0.0.2.0.0.0.1.0.0.0.0.0.0.0.1.2.3.4._ip6._fsv.example.com.

If an address is valid for a domain, its factored record exists and contains an A record with value 127.0.0.2. If its address is not valid, its factored record does not exist. Other values for an A record are reserved and MUST NOT be used. Wild cards can be used to decrease the number of unique records needed to encode a range of addresses. The addresses listed in the block and factored records MUST be the same.

For example, assume that example.com has outgoing mail servers in four CIDR ranges and a single additional address.

```
$ORIGIN example.com
; keep for a day
$TTL 24h
; the block record
; (the next line is broken in two for typographical reasons)
_fsv TXT "10.1.2.0/24" "10.3.4.0/23" "10.5.6.0/24"
             "10.7.8.8/30 "10.9.9.9"
; count of TXT fields, and also indicate that FSV data exists
_fsv A 0.0.0.5
; factored records
; 10.1.2/24
*.2.1.10._fsv A 127.0.0.2
; 10.3.4/23
*.4.3.10._fsv A 127.0.0.2
*.5.3.10._fsv A 127.0.0.2
; 10.5.6/24
*.6.5.10._fsv A 127.0.0.2
; 10.7.8.8/30
8.8.7.10._fsv A 127.0.0.2
9.8.7.10._fsv A 127.0.0.2
; 10.9.9.9
9.9.9.10._fsv A 127.0.0.2
```

Clients that use block data simply prepend _fsv to the domain and retrieve the TXT record. If none is available, the client may fetch the A record to check whether FSV data is supposed to be present and log an error if so. Once the TXT data is retrieved and parsed, subsequent mail from the same domain can be checked by looking up the incoming IP address in the data already retrieved. Clients MUST obey the TTL of any block data that is cached in an application and discard the data when the record expires.

Clients that use factored data construct the appropriate name by reversing the IP address, prepending it to _fsv and the domain name and fetching an A record. If the record exists and contains 127.0.0.2, the IP address is valid. If the

record does not exist, the client should fetch the base _fsv record to determine whether the domain publishes FSV data. Clients typically will not cache factored records, but if they do, they MUST obey the TTL of any data retrieved and discard

the data when the record expires,

7. Possible Extensions

The two high octets in the domain's _fsv could be used to publish flag bits about the domain's mail policy, for example, whether the domain permits roaming users to send mail through other hosts.

Other information about the domain's policy could be published in known subdomains of _fsv. For example, if it were desirable to publish a mail contact for a domain other than the standard Postmaster address, the e-mail address could be published in a TXT record in contact._fsv.domain.

8. Security Considerations

FSV data is only as secure as the DNS. DNS security is debatably inadequate (see [4]), but the fact that we've survived for 20 years suggests that it'll do for now. Should DNSSEC become available, it will apply to FSV data just like it does to the rest of the DNS.

If a domain's DNS servers are unavailable, due to local failures or a denial of service attack, SMTP recipients won't be able to validate mail from that domain. If FSV clients ``fail open'' and accept mail in the absence of FSV data, this would allow forged mail to be received. On the other hand, if FSV clients ``fail closed'' and reject mail temporarily or permanently in in the absence of FSV data, this would cause valid mail to be rejected. A possible design would be to return a temporary rejection in the absence of FSV data and hope that the FSV DNS failure would be cured before the message times out.

Block FSV data makes it easier to do security probes of a domain's authorized servers. An attacker might portscan all of the hosts in a domain's published FSV range, and if one of those hosts could be compromised, the attacker could then send forged mail with the domain's return address that passed FSV checks. Factored FSV data makes this attack somewhat harder.

9. Normative References

[1] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

10. Informative References

[2] Crocker, D. "Technical Considerations for Spam Control Mechanisms", work in progress, http://brandenburg.com/specifications/draft-crocker-spam-techconsider-02.txt

[3] Klensin, J. (Ed) "Simple Mail Transfer Protocol", RFC 2821, April 2001.

[4] D. Atkins and R. Austein, "Threat Analysis Of The Domain Name System," Internet Draft internet-drafts/draft-ietf-dnsext-dns-threats-05.txt, Nov 2003.

[5] P. Vixie, "Repudiating MAIL FROM", Internet Draft, June 6, 2002.

11. Author's Address

John R. Levine
Taughannock Networks
PO Box 727
Trumansburg NY 14886
(607) 330-5711
johnl@taugh.com