IETF 78 TPA-Label for ADSP

DKIM Third-Party Authorization Label
draft-otis-dkim-tpa-label

By Douglas Otis & Daniel Black

Acceptance w/o Author Domain Signature and ADSP

Reject or discard “discardable” w/o A-D sig breaks mailing lists

Reject “all” w/o A-D sig is not actionable

- Will reputation of signing domain's DKIM source compliance fix these issues?
- Will sender vouching of signing domains for handling of sources and third-party services for handling of A-R header fields fix these issues?
- Perhaps informal third-party services are used so rarely, that exceptions can be manually maintained by each recipient?

Effective Phishing Mitigation

More than 20k domains just within the financial sector are actively being targeted by phishing, which thwarts manual tactics.

Increase in the number of subdomains being used shifts the problem, adds to mitigation efforts, without fully containing the phishing threat. With this, perhaps ADSP ends up making phishing worse.

Reputation services can not reasonably attest to a domain's DKIM compliance, nor whether informal services are ever used.

Since an Author Domain's email is at risk, only it should indicate whether informal third-party services are used, or whether these services properly handle A-R header fields. To allow sorting, in some cases additional header fields need to be stipulated.

Informal TPA-Label Authorization

- Offers protective practices without “discardable”
- Avoids unprotected subdomain or additional domain use
- Leverages other authentication methods
- Leverages ISP with DKIM lacking protective practices
- Protects phished domains use of informal services
- Senders then map acceptable third-party services
- Makes no Identity claims
- Dramatically and proactively reduces avenues of attack

Scope Method Breakdown

scope values	Field or Parameter	Method
F	From (Author) Header Field	Match Address Domain
L	List-ID Header Field	Match List-ID Identifier
S	Sender Header Field	Match Address Domain
e	SMTP Hostname	Resolve Hostname IP Addr
h	SMTP Hostname	Pass SPF with Hostname
m	MailFrom	Pass SPF with MailFrom
t	SMTP Hostname	Cert of Hostname

Cryptographic Path Check

Currently, no general practice employs certificates to confirm the domain of the client initiating a connection. This may be needed for clients within IPv6 IP address space where tunneling, carrier grade NATs, and rapid space assignment without any practical reverse mapping, reduces effectiveness of IP address based reputations.

There is an existing TLS option for SMTP and an ongoing effort to standardize automated server confirmation. It might be possible to leverage this effort to establish practices used at the client. For information related to ongoing server related efforts see: http://tools.ietf.org/html/draft-saintandre-tls-server-id-check-08

It might also be possible to utilize the DKIM public key to verify a challenge signed by the client based upon keys located at its hostname, but this would require a change made to SMTP conversations defined in RFC4954 Section 4.

DKIM ABNF for list

```
tag-list  =  tag-spec 0*( ";" tag-spec ) [ ";" ]

tag-spec  =  [FWS] tag-name [FWS] "=" [FWS] tag-value [FWS]

tag-name  =  ALPHA 0*ALNUMPUNC

tag-value =  [ tval 0*( 1*(WSP / FWS) tval ) ]

; WSP and FWS prohibited at beginning and end
  tval      =  1*VALCHAR

  VALCHAR   =  %x21-3A / %x3C-7E
; EXCLAMATION to TILDE except SEMICOLON
  ALNUMPUNC =  ALPHA / DIGIT / "_"
```

ADSP ABNF for “dkim” tag

```
adsp-dkim-tag = %x64.6b.69.6d *WSP "=" *WSP
                ("unknown" / "all" / "discardable" / x-adsp-dkim-tag)

x-adsp-dkim-tag = hyphenated-word    ; for future extension
; hyphenated-word is defined in RFC 4871
```

TPA-Label dkim tag values

dkim values	Field or Parameter
tpa-sig	Third-Party Signature
tpa-path	Third-Party Path or Signature
all tpa-sig	Third-Party Signature
discardable	Selective discardable

Label Generation Definition

(underscore) base32(sha1(lcase(tpa-domain)))

The label encoding process inputs the hash as a byte stream of four 40-bit data blocks where each data block outputs 8 encoded characters. Proceeding from left to right, a 40-bit input group is formed by concatenating 5 bytes. The 40-bit input is then treated as 8 concatenated 5-bit groups, each of which is translated into a single digit of the base32 alphabet. The bit stream is ordered with the most-significant-bit first, being the high-order bit of the first byte. The entire output is then concatenated first to last, left to right, into 32 characters prefixed with an underscore.