DSAP
DKIM Signature Authorization Protocol

Hector Santos
Santronics Software, Inc.
DSAP Summary

- Security Problem:
  - DKIM-BASE is an unprotected mail authentication and identification protocol.

- DSAP Solution:
  - Provide simple to implement DNS-based robust security wrapper to secure the unprotected DKIM-BASE protocol.
  - Provide consistent protocol support software designs.
DSAP Goal and Objective

- Protects Domain DKIM message signing Practice.
- Protects Domain Reputations.
- Reduces DKIM Verification Overhead.
- Simplifies DKIM Implementation Design considerations.
- Increases DKIM acceptability and lowers Adoptions Barriers
Unprotected DKIM Protocol

- Intentional vague semantics.
- No protection against domain name exploitations.
- No foundation for consistent DKIM verification.
- Increases verification overhead.
- Places high burden on verification receivers.
- Little payoff (low efficiency).
- Hedges future on unknown, yet to be delivered, trusted-layers protocols (Reputation Services).
How Did We Get Here?

- Original DKIM proof of concept included SSP (Sender Signing Policies).
- Separation of DKIM and SSP protocol.
- Poor SSP functional specifications.
- SSP de-emphasized in lieu of future trusted-layers business ventures.
- Making DKIM-BASE a standalone and unprotected protocol.
Other Non-SSP Considerations:

- **Trusted-Layers - Reputation Services**
  - No Standard
  - 3\textsuperscript{rd} party Trust Required
  - "Batteries Required" Dilemma
  - Highly isolated solution.

- **LMAP Solutions**
  - SMTP based
  - Probably will be augmented as part of solution.

**Problem?**
None offer direct protection for DKIM Signature
DKIM without DSAP

- **PAYLOAD**
- **CONTINUE**

**UNSIGNED MESSAGE**

**SIGNED MESSAGE**

**SIGNATURE**

**INVALID SIGNATURE**

**VALID SIGNATURE**

**PASS**

**CONTINUE/CLASSIFY**

**FAIL/CLASSIFY**

**UNDETECTED FAILURE AND FRAUD**

DKIM recommends to view invalid signatures as if it didn’t exist.
Fundamental Flaws

- Accept (Pass) Valid DKIM signatures
- Ignore Invalid DKIM signatures
- Void of Highly Detectable Failures
DKIM with DSAP:

- **Payload**
- **CONTINUE**
- **UNSIGNED/SIGNATURE OPTIONAL POLICIES**
- **NO MAIL POLICY**
- **UNSIGNED/EXCLUSIVE/STRONG POLICIES**
- **SIGNED/NONE/NXDOMAIN POLICY**
- **FAIL/CLASSIFY**
- **3PS SIGNED/NO 3PS POLICIES**
- **CONTINUE/CLASSIFY**
- **VALID SIGNATURE**
- **INVALID SIGNATURE**
- **PASS**
- **DKIM**
- **SSP**

DKIM-BASE recommends to view invalid signatures as if it didn't exist.
## Detectable Failures

<table>
<thead>
<tr>
<th>DKIM</th>
<th>SIGNING PRACTICE</th>
</tr>
</thead>
<tbody>
<tr>
<td>UNSIGNED</td>
<td>EXPECTED (Exclusive, Strong)</td>
</tr>
<tr>
<td>SIGNED</td>
<td>NOT EXPECTED (None, NXDOMAIN)</td>
</tr>
<tr>
<td>SIGNED</td>
<td>NO 3(^{rd}) PARTY (Exclusive, Weak)</td>
</tr>
</tbody>
</table>
Non-Detectable Failures

- Altered Message Body Integrity
- Reordering of RFC 2822 headers
Why not use SSP?

- Concerns about additional DNS lookups.
- Incomplete Protection.
- Incorrect DKIM integration.
- Not well understood (because of flaws).
- No consensus (because of flaws).
Two DNS records
- DSAP Policy Record
- Public DKIM Key Record

Two Maximize Lookups
- Policy: _selector._dkim.domain.com
- Key: _dkim.domain.com

With DSAP, Policy can short circuit Key lookup minimizing additional lookup concerns.
## Current SSP Policies:

<table>
<thead>
<tr>
<th>SSP Policy</th>
<th>Declaration</th>
</tr>
</thead>
<tbody>
<tr>
<td>NXDOMAIN</td>
<td>No SSP record defaults to NEUTRAL</td>
</tr>
<tr>
<td>NOMAIL (0=.)</td>
<td>No Mail Expected</td>
</tr>
<tr>
<td>NONE (undefined)</td>
<td>No Signature Expected</td>
</tr>
<tr>
<td>WEAK (o=? proposed)</td>
<td>Signature Optional, No 3PS</td>
</tr>
<tr>
<td>NEUTRAL (o=~)</td>
<td>Signature Optional, 3PS allowed</td>
</tr>
<tr>
<td>STRONG (o=-)</td>
<td>Signature Expected, 3PS allowed</td>
</tr>
<tr>
<td>EXCLUSIVE (o=!</td>
<td>Signature Expected, No 3PS</td>
</tr>
<tr>
<td>USER (o=^)</td>
<td>Signature Expected</td>
</tr>
</tbody>
</table>
DSAP - Verifier Viewpoint:

- Original Party Signature (OPS)
  - Not Expected (-)
  - Expected (+)
  - Optional (~)
- 3rd Party Signature (3PS)
  - No Expected (-)
  - Expected (+)
  - Optional (~)
## Possible OPS and 3PS Policies

<table>
<thead>
<tr>
<th>OPS</th>
<th>3PS</th>
<th>SSP (o=)</th>
<th>DSAP (sp=)</th>
</tr>
</thead>
<tbody>
<tr>
<td>NO MAIL</td>
<td>NOMAIL</td>
<td>SP=;</td>
<td></td>
</tr>
<tr>
<td>NOT EXPECTED</td>
<td>NOT EXPECTED</td>
<td>NONE</td>
<td>OP-,3P-</td>
</tr>
<tr>
<td>NOT EXPECTED</td>
<td>EXPECTED</td>
<td>UNDEFINED</td>
<td>OP-,3P+</td>
</tr>
<tr>
<td>NOT EXPECTED</td>
<td>OPTIONAL</td>
<td>UNDEFINED</td>
<td>OP-,3P~</td>
</tr>
<tr>
<td>EXPECTED</td>
<td>NOT EXPECTED</td>
<td>EXCLUSIVE</td>
<td>OP+,3P-</td>
</tr>
<tr>
<td>EXPECTED</td>
<td>EXPECTED</td>
<td>UNDEFINED</td>
<td>OP+,3P+</td>
</tr>
<tr>
<td>EXPECTED</td>
<td>OPTIONAL</td>
<td>STRONG</td>
<td>OP+,3P~</td>
</tr>
<tr>
<td>OPTIONAL</td>
<td>NOT EXPECTED</td>
<td>WEAK</td>
<td>OP~,3P-</td>
</tr>
<tr>
<td>OPTIONAL</td>
<td>EXPECTED</td>
<td>UNDEFINED</td>
<td>OP~,3P+</td>
</tr>
<tr>
<td>OPTIONAL</td>
<td>OPTIONAL</td>
<td>NEUTRAL</td>
<td>OP~,3P~</td>
</tr>
</tbody>
</table>
Multiple Signatures:

- Policies allows 3\textsuperscript{rd} Party Signatures (3PS).
  - OP+,3P+
  - OP+,3P~ (SSP, o=STRONG)
  - OP~,3P+
  - OP~,3P+

- Reasons for 3PS (or re-signers).
  - Broken Integrity
  - Vendor Relationships (ISP, EPS, Clearinghouse)
  - Middleware requirements

- Original domains need to decide if multiple signatures are acceptable. If not, declare a 3P- policy.

- Domains with signature requirements but allow middleware changes should declare a strong resigning requirement policy (OP+, 3P~).
Middle Ware & List Servers:

- Identify middle ware design change requirements.
- Problem remains with LS integrity changes.
- Regulate Subscription from Restrictive DSAP Policies.
- Use DSAP policies to determine and honor 1\textsuperscript{st} party versus 3\textsuperscript{rd} party signature requirements.
Recommendation

- Domains should not expose their domain reputation with a DKIM-BASE only implementation.
- Implement DSAP with DKIM-BASE.
- Analyze Domain Usage for proper DSAP policy declarations.
What’s Next?

- Obtain WG feedback,
- Assist Developers with cross platform implementation DSAP models.
Conclusion

In order for DKIM to be well accepted, it needs to offer value to all parties.

DSAP adds a simple to implement security layer around the unprotected core DKIM protocol.

DKIM should be a fundamental natural part of DKIM protocol.

If implemented, DKIM will have less of a negative impact on verifiers and also make it easier for developers to add DKIM signing support.
Hector Santos

- hsantos@santronics.com
- http://www.santronics.com
- Wildcat! Interactive Net Server
- Silver Xpress Mail System