Best Practice for Email Authentication - Optimal Ways to Deploy SPF, DKIM and DMARC

Contents

Introduction

This guide describes three predominant email authentication technologies in use today - SPF, DKIM, and DMARC, and discusses various aspects of their implementation. Several real-life email architecture situations are discussed, and guidelines for implementing them on the Cisco Email Security product set. Since this is a hands-on best practices guide, some of the more complex material will be omitted. When necessary, certain concepts may be simplified or condensed to ease understanding of the presented matter.

Product knowledge requirements

This guide is an advanced level document. To follow through with the material presented, the reader should possess product knowledge of the Cisco Email Security Appliance to the level of Cisco Email Security Field Engineer certification. Furthermore, readers should have a strong command of DNS and SMTP and their operation. Acquaintance with the basics of SPF, DKIM, and DMARC is a plus.

Email Authentication – A Short Overview

Sender Policy Framework (SPF)

Sender Policy Framework was first published in 2006, as RFC4408. The current version is specified in RFC7208 and updated in RFC7372. In essence, it provides a simple way for a Domain Owner to advertise their legitimate email sources to the Receivers using DNS. Although SPF primarily authenticates the return path (MAIL FROM) address, the specification recommends (and provides mechanism) to also authenticate SMTP HELO/EHLO argument (FQDN of sender's gateway as transmitted during SMTP conversation).

SPF uses TXT type DNS Resource Records of fairly simple syntax:

```
spirit.com        text = "v=spf1 mx a ip4:38.103.84.0/24 a:mx3.spirit.com
a:mx4.spirit.com include:spf.protection.outlook.com ~all"
```

The Spirit Airlines record above allows email from @spirit.com addresses to come from a particular /24 subnet, two machines identified by a FQDN, and Microsoft's Office365 environment. The "~all" qualifier at the end instructs receivers to consider any other sources as Soft Fail – one of two failure modes of SPF. Take note that senders do not specify what receivers should do with failing messages, just to which degree they will fail.

Delta, on the other hand, employs a different SPF scheme:

```
delta.com        text = "v=spf1 a:smtp.hosts.delta.com
include:_spf.vendor.delta.com -all"
```

To minimize the number of DNS queries required, Delta created a single "A" record listing all of its SMTP gateways. They also provide a separate SPF record for their vendors in "_spf.vendor.delta.com". They also include instructions to **Hard Fail** any messages not authenticated by SPF ("-all" qualifier). We can further look up the vendors' SPF record:

```
_spf.vendor.delta.com  text = "v=spf1 include:_spf-delta.vrli.com
include:_spf-ncr.delta.com a:delta-spf.niceondemand.com
include:_spf.airfrance.fr include:_spf.qemailserver.com
include:skytel.com include:epsl1.com ?all"
```

So, emails from senders @delta.com may legitimately come from, for example, Air France's email gateways.

United, on the other hand, uses a much simpler SPF scheme:

```
united.com      text = "v=spf1 include:spf.enviaremails.com.br
include:spf.usa.net include:coair.com ip4:161.215.0.0/16
ip4:209.87.112.0/20 ip4:74.112.71.93 ip4:74.209.251.0/24 mx ~all"
```

Other than their own corporate mail gateways, they include their email marketing providers (“usa.net” and “enviaremails.com.br”), legacy Continental Air Lines gateways, as well as everything listed in their MX records (“MX” mechanism). Take note that MX (an **incoming** mail gateway for a domain) may not be the same as **outgoing**. While for smaller enterprises they will usually be the same, larger organizations will have separate infrastructure handling incoming mail, and separate handling outgoing delivery.

Also, worth noting is that all of the above examples make extensive use of additional DNS referrals (“include” mechanisms). However, due to performance reasons, SPF specification limits the total number of DNS lookups necessary to retrieve a final record to **ten**. Any SPF lookups with over 10 levels of DNS recursion will fail.

Domain Keys Identified Mail (DKIM)

DKIM, specified in RFCs 5585, 6376 and 5863 is a merge of two historic proposals: Yahoo’s DomainKeys and Cisco’s Identified Internet Mail. It provides a simple way for senders to cryptographically sign outgoing messages and include the signatures (along with other verification metadata) in an email header (“DKIM-Signature”). Senders publish their public key in the DNS, thus making it easy for any receivers to retrieve the key and verify signatures. DKIM does not authenticate the source of the physical messages but relies on the fact that if the source is in possession of the sender organization’s private key, it is implicitly authorized to send an email on their behalf.

To implement DKIM, sending organization would generate one or more public key pairs and publish the public keys in the DNS as TXT records. Each key pair would be referenced by a “selector” so DKIM verifiers can differentiate between keys. Outgoing messages would be signed, and DKIM-Signature header inserted:

```
DKIM-Signature: v=1; a=rsa-sha1; c=relaxed/relaxed; s=united;
d=news.united.com;h=MIME-Version:Content-Type:Content-Transfer-
Encoding:Date:To:From:Reply-To:Subject:List-Unsubscribe:Message-ID;
i=MileagePlus@news.united.com; bh=IBSWR4yzI1PSRYtWLx4SRDSWII4=;

b=HrN5QINgnXwqkx+Zc/9VZys+yhikrP6wSZVu35KA0jfgYzhzSdfA2nA8D2JYIFTNLO8j4D
GmKhH1MMTyYgwYqT01rEwL0V8MEY1MzxTrzijkLPGqt/sK1WZt9pBacEw1fMWRQLf3BxZ3ja
YtLoJMRwxtgoWdfHU35CsFG2CNYLo=
```

The format of the signature is fairly straightforward. “a” tag specifies algorithms used for signing, “c” specifies the canonicalization scheme(s) used [1], “s” is the selector or key reference, “d” is the signing domain. The rest of this DKIM-Signature header is message specific: “h” lists signed headers, “i” lists signing user’s identity, and finally the header ends with two separate hashes: “bh” is a hash of signed headers, while “b” is the hash value for the body of the message.

When receiving a DKIM-signed message, the receiver will look up the public key by constructing the following DNS query:

```
<selector>._domainkey.<signing domain>
```

as specified in the DKIM-Signature header. For the above example, our query would be “united._domainkey.news.united.com”:

```
united._domainkey.news.united.com  text = "g=*\; k=rsa\; n=" "Contact"
```

```
"postmaster@responsys.com" "with" "any" "questions" "concerning" "this"
"signing" "\;
p=MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQC/Vh/xq+sSRLhL5CRU1drFTGMXX/Q2Kk
Wgl35hO4v6dTy5Qmxcuv5AwqxLiz9d0jBaxtuvYALjlGkxmk5MemgAOcCr97GlW7Cr11eLn8
7qdTmyE5LevnTXxVDMjIfQJt6OFzmw6Tp1t05NPWh0PbyUohZYt4qpcbiz9Kc3UB2IBwIDAQ
AB\;"
```

DNS record returned contains the key, as well as other optional parameters. [2]

The main problem with DKIM is that the initial specification did not allow for advertising that a sender uses DKIM. Thus, if a message comes without a signature, there is no easy way for a receiver to know that it should have been signed and that in that case, it's most probably not authentic. Since a single organization can (and most often will) use several selectors, it is not trivial to "guess" whether a domain is DKIM-enabled. A separate standard, Author Domain Signing Practices, was developed to cover this, but due to low use and other issues was obsoleted in 2013 with no successor.

Domain-based Message Authentication, Reporting And Conformance (DMARC)

DMARC is the youngest of the three email authentication technologies covered and was developed specifically to address the shortcomings of both SPF and DKIM. Unlike the other two, it authenticates the Header From of a message and links into the checks previously performed by the other two. DMARC is specified in RFC7489.

Added-value of DMARC over SPF and DKIM comprises of:

- Making sure that all available identities (HELO, MAIL FROM and/or DKIM signing domain) are aligned (exactly matching or subordinate) with From header
- Providing a means for the sender domain owner to specify a policy for receivers on how they **must** handle failing messages
- Providing a feedback facility for sender domain owners to be informed of any failing messages thus making it easy to identify phishing campaigns or errors in SPF/DKIM/DMARC policy assignment

DMARC also uses a simple DNS-based policy distribution mechanism:

```
_dmarc.aa.com     text = "v=DMARC1\; p=none\; fo=1\; ri=3600\;
rua=mailto:american@rua.agari.com,mailto:dmarc@aa.com\;
ruf=mailto:american@ruf.agari.com,mailto:dmarc@aa.com"
```

The only mandatory tag in DMARC policy specification is "p", specifying the policy to use on failing messages. It can be one of the three: none, quarantine, reject.

Most often used optional parameters have to do with reporting: "rua" specifies a URL (either a mailto: or an http:// URL using POST method) to send daily aggregate reports on all failing messages purporting to come from a particular domain. "ruf" specifies a URL to submit immediate detailed failure reports on every failing message.

According to specification, a receiver **must** adhere to the advertised policy. If they do not, they **must** notify the sender domain owner in the aggregate report.

The central concept of DMARC is the so-called identifier alignment. Identifier alignment defines how a message can pass DMARC verification. SPF and DKIM identifiers are aligned separately, and a message needs to pass **any** of them to pass DMARC overall. However, there is a DMARC policy option where the sender can request that a failure report be generated even if one alignment passes, but the other fails. We can see this in the above example with "fo" tag being set to "1".

There are two ways for messages to adhere to either DKIM or SPF identifier alignment, strict and relaxed. Strict adherence means that FQDN of Header From must fully match the Signing Domain ID ("d" tag) of DKIM signature or FQDN of MAIL FROM SMTP command for SPF. Relaxed, on the other hand, allows Header From FQDN to be a subdomain of the fore mentioned two. This has important implications when delegating your email traffic to third-parties, which will be discussed later in the document.

SPF Deployment Considerations

SPF For Receivers

SPF verification is trivial to configure on the Cisco Email Security Appliance or Cloud Email Security virtual appliances. For the remainder of this document, any reference to ESA will also include CES.

SPF verification is configured in Mail Flow Policies – the easiest way to run it globally is to turn it on in the Default Policy Parameters section of the appropriate listener(s). If you are using the same listener for incoming and outgoing mail collection, make sure that your "RELAYED" Mail Flow Policy has SPF verification set to "Off".

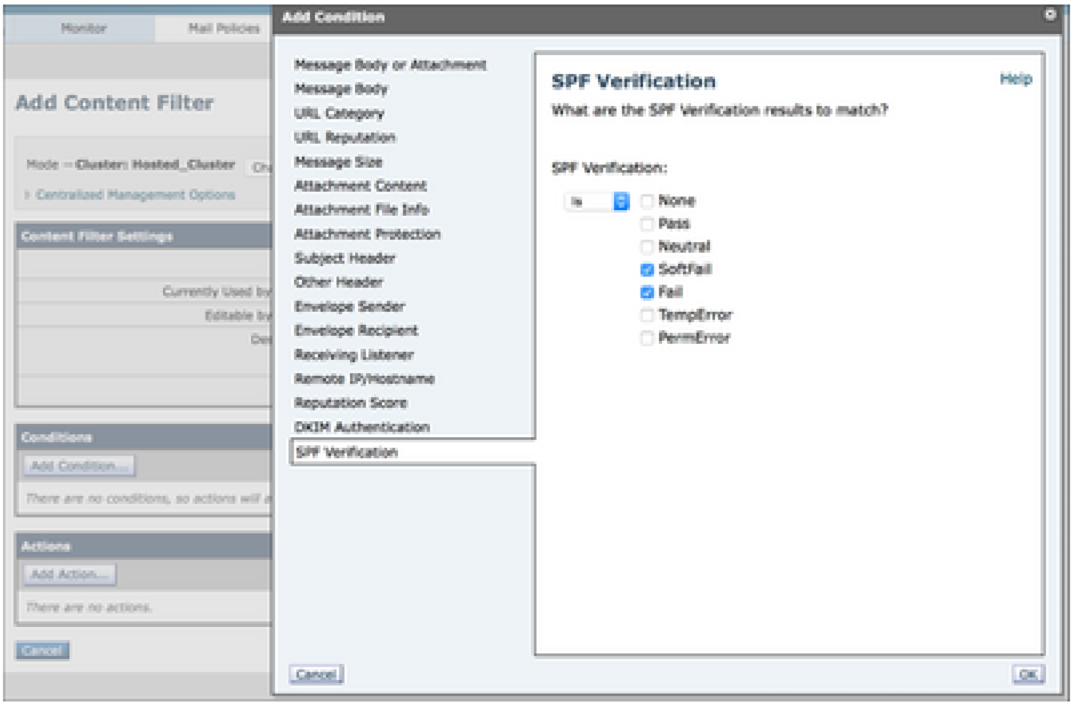
Since SPF does not allow for specification of policy action to be taken, SPF verification (as well as DKIM, as we shall see later on) only verifies the message and inserts a set of headers for each SPF check performed:

```
Received-SPF: Pass (mx1.hc4-93.c3s2.smtpi.com: domain of

  united.5765@envfrm.rsys2.com designates 12.130.136.195 as

  permitted sender) identity=mailfrom;

  client-ip=12.130.136.195; receiver=mx1.hc4-93.c3s2.smtpi.com;

  envelope-from="united.5765@envfrm.rsys2.com";

  x-sender="united.5765@envfrm.rsys2.com";

  x-conformance=sidf_compatible; x-record-type="v=spf1"

Received-SPF: None (mx1.hc4-93.c3s2.smtpi.com: no sender

  authenticity information available from domain of

  postmaster@omp.news.united.com) identity=helo;

  client-ip=12.130.136.195; receiver=mx1.hc4-93.c3s2.smtpi.com;
  envelope-from="united.5765@envfrm.rsys2.com";

  x-sender="postmaster@omp.news.united.com";

  x-conformance=sidf_compatible
```

Take note that for this message, two “identities” were verified by SPF: “mailfrom” as mandated by the specification, and “helo” as recommended by the same. The message will formally pass SPF, since only the former is relevant for SPF compliance, but some receivers may sanction senders that don’t include SPF records for their HELO identities as well. Therefore, it is good practice to include your outgoing mail gateways’ hostnames in your SPF records.

Once Mail Flow Policies verify a message, it is up to local administrators to configure an action to be taken. This is done using Message Filter rule SPF-status() [3], or by creating an Incoming Content Filter using the same and applying it to appropriate Incoming Mail Policies.

Picture 1: SPF Verification Content Filter Condition



SPF Verification

What are the SPF Verification results to match?

SPF Verification:

Is
□ None
□ Pass
□ Neutral
☑ SoftFail
☑ Fail
□ TempError
□ PermError

Recommended filter actions are to drop messages that Fail (“-all” in the SPF record), and quarantine messages that Softfail (“~all” in the SPF record) in a Policy Quarantine, however, this might vary according to your security requirements. Some receivers just tag failing messages, or take no visible action, but report it to the administrators.

Recently there has been a significant surge in SPF popularity, but many domains publish incomplete or incorrect SPF records. To be on the safe side, you may want to quarantine all SPF-failing messages, and monitor the quarantine for a while, to make sure there are no “false positives”.

If You Provide Email Services For Other Domains Or Third-Parties

If you provide email delivery or hosting services for third-parties, they will have to add hostnames and IP addresses you use to deliver their messages to their own SPF records. The easiest way to do this is for the provider to create an “umbrella” SPF record, and have customers use “include” mechanism in their SPF records.

```
suncountry.com    text = "v=spf1 mx ip4:207.238.249.242
ip4:146.88.177.148 ip4:146.88.177.149 ip4:67.109.66.68
ip4:198.179.134.238 ip4:107.20.247.57 ip4:207.87.182.66
ip4:199.66.248.0/22 include:cust-spf.exacttarget.com ~all"
```

As we can see, Sun Country has some of their emails under their own control, but their marketing email is outsourced to a third-party. Expanding the referred record reveals a list of current IP addresses used by their marketing mailing service provider:

```
cust-spf.exacttarget.com      text = " v=spf1 ip4:64.132.92.0/24
ip4:64.132.88.0/23 ip4:66.231.80.0/20 ip4:68.232.192.0/20
ip4:199.122.120.0/21 ip4:207.67.38.0/24 ip4:207.67.98.192/27
ip4:207.250.68.0/24 ip4:209.43.22.0/28 ip4:198.245.80.0/20
ip4:136.147.128.0/20 ip4:136.147.176.0/20 ip4:13.111.0.0/18 -all"
```

This flexibility allows email service providers to scale without having to reach out to each customer to modify their DNS records.

If You Use Third-Party Email Services

Similarly to the previous paragraph, if you are using any third-party emailing services, and wish to establish fully SPF-verified mail flow, you must include their own SPF records in yours.

```
jetblue.com descriptive text "v=spf1 include:_spf.qualtrics.com ?all"
```

JetBlue uses Qualtrics analytics service, and the only thing they needed to do is include a correct SPF record from Qualtrics. Similarly, most other ESPs provide SPF records to be included in their customers' records.

If your ESP or email marketer doesn't provide SPF records, you will have to list their outgoing mail gateways directly in yours. However, it is then your responsibility to keep those records accurate, and if the provider adds additional gateways or changes IP addresses or hostnames, your mail flow may be jeopardized.

Additional danger from third-parties who aren't SPF-conscious comes from sharing resources: If an ESP uses the same IP address to deliver email of several customers, it is technically possible for one customer to generate SPF-valid message pretending to be another customer delivering through the same interface. This is why, before putting in place any SPF restrictions, you should investigate your MSP's security policies and awareness of email authentication. If they don't have answers to your questions, considering how SPF is one of the basic mechanisms of trust on the Internet, you are strongly advised to reconsider your choice of MSP. It's not only about security – SPF, DKIM, DMARC and other senders best practices [4]employed by MSPs are an assurance of deliverability. If your MSP does not follow them or follows them incorrectly, that will lower their trustworthiness with large receiving systems and possibly delay or even block your messages.

(Sub)Domains with No Email Traffic

Most organizations today own several domains for marketing purposes but only use one actively for corporate email traffic. Even if SPF is correctly deployed on the production domain, bad actors can still use other domains that are not actively used for an email to spoof an organization's identity. SPF can prevent this from occurring through a special "deny all" SPF record – for any of your domains (and subdomains!) that don't generate email traffic, publish "v=spf1 –all" in the DNS.

An excellent example is openspf.org – the Website of the SPF Council.

Since SPF delegation is valid only for a single domain, it is critical to also publish “deny all” SPF records for any subdomains you may be using that might not generate an email. Even if your production domain has a “regular” SPF record, do make an extra effort to add “deny all” records to your subdomains with no traffic. And again – don’t forget that receiving is not equivalent to sending: A domain may very well be receiving email, but will never be a source. This is very true for short-term marketing domains (e.g. events, limited time promotions, product launches…), where emails incoming to those domains would be delivered to your production domain, and any responses to those emails will be delivered from the production domain. These short-term domains will have a valid MX record but should have an SPF record that identifies them as no **source** of email as well.

DKIM Deployment Considerations

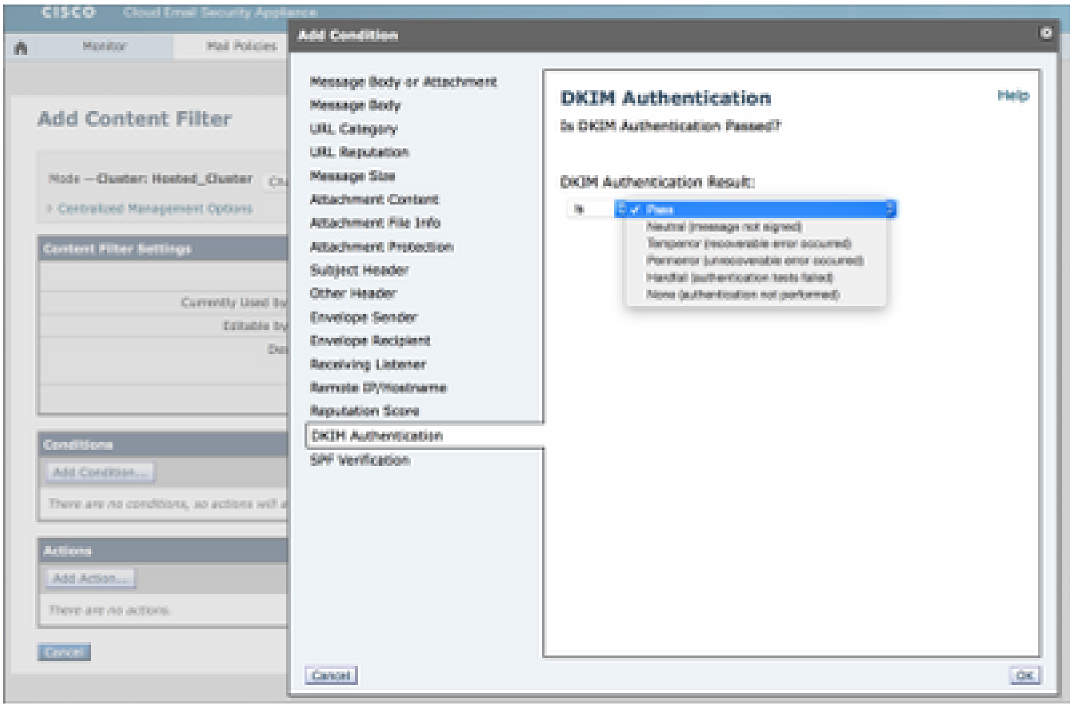
DKIM For Receivers

Configuring DKIM verification on the ESA is similar to SPF verification. In the Default Policy Parameters of Mail Flow Policies, simply turn DKIM Verification to “On”. Again, since DKIM does not allow for any policy specification, this will just verify the signature and insert an “Authentication-Results” header:

```
Authentication-Results: mx1.hc4-93.c3s2.smtpi.com; dkim=pass (signature
verified) header.i=MileagePlus@news.united.com
```

Any actions based on DKIM verification results have to be performed by Content Filters:

Picture 2: DKIM Verification Content Filter condition



DKIM Authentication

Is DKIM Authentication Passed?

DKIM Authentication Result: is ✓ Pass

Unlike SPF, which is straightforward, DKIM manipulates the actual message text, so some parameters may be limited. Optionally, you can create DKIM Verification Profiles, and assign different Verification Profiles to different Mail Flow Policies. They allow you to limit key sizes of signatures you will accept, set key retrieval failure actions and configure the depth of DKIM verification.

As a message passes multiple gateways, it can be signed multiple times and thus carry multiple signatures. For a message to pass DKIM Verification, **any** signatures need to verify. By default, ESA will verify up to five signatures.

Due to historic openness of SMTP and email and the reluctance of overall Internet to adapt to (positive) changes, there are still several situations when DKIM signatures might legitimately fail such as when mailing list managers directly relay but modify messages, or when messages are forwarded directly rather than as attachments to new messages. This is why, in general, best practice for messages failing DKIM would still be to quarantine or tag, rather than drop them.

Preparing to sign with DKIM

Before you can turn on DKIM Signing in your RELAYED Mail Flow Policy, you need to generate/import the keys, create DKIM Signing Profile(s) and publish the public key(s) in the DNS.

If you are signing for a single domain, the process is straightforward. Generate the key pair, create your single Signing Profile in the Domain Keys section of Mail Policies, and click the “Generate” option under “DNS Text Record” once your profile is ready. Publish the key as generated in your DNS. Finally, turn on DKIM Signing in your Mail Flow Policy.

It gets more complicated if you’re signing for several distinct domains. In that case, you have two options:

1. Use a single Signing Profile to sign for all domains. You will store the (single) public key in the DNS zone of the “primary” domain, and your DKIM Signatures will reference that key. This technique was often employed by ESPs in the past – it allowed them to sign on a large scale, while not having to interact with individual customers’ DNS space [5].
2. Create a separate Signing Profile for each domain you sign for. This makes for more complex initial configuration but provides much more flexibility moving forward. Create a key pair for each domain, create a profile specifying only one domain (and its subdomains) in the “Profile Users” section, and publish the relevant public key in that particular domain’s DNS zone.

Although option #1 is easier to start with, remember that it will ultimately break DMARC. Since DMARC requires that Signing Domain ID be aligned with Header From, your identifier alignment with DKIM will fail. You may be able to get away with it if you configure your SPF correctly, and rely on SPF identifier alignment to pass DMARC verification.

However, by implementing option #2 from the start, you don’t need to worry about DMARC and it is pretty easy to revoke or reconfigure signing service for just a single domain. Also, if you provide **some** email services for a third-party domain, you will most probably need to get the key to use from them (and import it into your ESA). That key will be domain-specific, thus you will need to create a separate profile.

If You Use Third-Party Email Services

In general, if you use DKIM signing and offload some of your email processing (e.g. marketing emails) to a third-party, you don’t want them to use the same keys that you use in production. This is one of the main reasons for the existence of Selectors in DKIM. Instead, you should generate a new key pair, publish the public portion in your DNS zone and deliver the secret key to the other party. This will also allow you to quickly revoke that particular key in case of trouble while keeping your production DKIM infrastructure untouched.

Although it is not necessary for DKIM (messages for the same domain can be signed with multiple different keys), it is good practice to provide a separate subdomain for any email that is handled by a third-party. It will make tracking the messages easier, and will allow for much cleaner implementation of DMARC later on. As an example, consider these five DKIM-Signature headers from multiple messages from Lufthansa:

```
DKIM-Signature: v=1; a=rsa-sha1; c=relaxed/relaxed; s=lufthansa;
d=newsletter.milesandmore.com;

DKIM-Signature: v=1; a=rsa-sha1; c=relaxed/relaxed; s=lufthansa2;
d=newsletter.lufthansa.com;

DKIM-Signature: v=1; a=rsa-sha1; c=relaxed/relaxed; s=lufthansa3;
d=lh.lufthansa.com;

DKIM-Signature: v=1; a=rsa-sha1; c=relaxed/relaxed; s=lufthansa4;
d=e.milesandmore.com

DKIM-Signature: v=1; a=rsa-sha1; c=relaxed/relaxed; s=lufthansa5; d=fly-
lh.lufthansa.com;
```

We can see that Lufthansa is using five different keys (selectors) split over five separate subdomains of two primary production domains (lufthansa.com and milesandmore.com). This means that each of these can be independently controlled, and each can be outsourced to a different messaging service provider.

DMARC Deployment Considerations

DMARC For Receivers

DMARC verification on the ESA is profile-based, but unlike DKIM, the Default profile must be edited to be compliant with the specification. The default behavior of the ESA is to never drop any messages unless explicitly instructed by the customer, so default DMARC verification profile will have all actions set to “No Action”. Additionally, to enable correct report generation, you will need to edit "Global Settings" of the DMARC section of "Mail Policies".

Once a profile has been set up, DMARC verification, just like the other two, is set in the Default Policy Settings section of Mail Flow Policies. Make sure to check the box to send aggregate feedback reports – this is arguably the most important feature of DMARC for the sender. At the time of writing, ESA does not support the generation of per-message failure reports (“ruf” tag of DMARC policy).

As DMARC policy actions are advised by the sender, unlike SPF or DKIM, there are no specific actions configurable outside of profile configuration. It’s not necessary to create any content filters.

DMARC verification will add additional fields to the Authentication-Results header:

```
Authentication-Results: mx1.hc4-93.c3s2.smtpi.com; dkim=pass (signature
verified) header.i=MileagePlus@news.united.com; dmarc=pass (p=none
dis=none) d=news.united.com
```

In the above example, we see that DMARC was verified based on DKIM identifier alignment, and sender requested policy of “none”. This indicates that they are currently in the “monitor” phase of DMARC deployment.

If You Provide Email Services For Other Domains Or Third-Parties

The biggest concern of ESPs for DMARC compliance is to achieve proper identifier alignment. When planning for DMARC, make sure that your SPF is set up correctly, that all relevant other domains have your outgoing gateways in your SPF records and that they don’t submit messages that will fail alignment, primarily by using different domains for MAIL FROM and Header From identity. This error is most often done by applications that send email notifications or warnings because application authors are mostly unaware of the consequences of the inconsistency of their email identities.

As described previously, make sure that you use a separate DKIM signing profile for each domain, and that your signing profile properly references the domain you are signing for as used in Header From. If you are using your own subdomains, you **can** sign with a single key, but make sure you set your adherence to DKIM to relaxed in the DMARC policy (“adkim=”r”).

In general, if you are providing email services for a larger number of third-parties that you don’t have direct control over, it is good practice to write a guideline document on how to submit an email that is most likely to deliver. As user-to-user email is generally well behaved, this will mostly serve as a policy document for application authors in the examples mentioned above.

If You Use Third-Party Email Services

If you use third-parties to deliver some of your email traffic, the optimal way is to delegate a separate subdomain (or a completely different domain) to the third-party provider. This way they can manage the SPF records as needed, have separate DKIM signing infrastructure, and not interfere with your production traffic. Then, DMARC policy for outsourced email can be different than for in-house. As already mentioned, when considering the third-party delivered email, always make sure that your identifiers will align, and your adherence to DKIM and SPF is set appropriately in your DMARC policy.

(Sub)Domains with No Email Traffic

Another improvement of DMARC over previous email authentication technologies is how it handles subdomains. By default, a particular domain’s DMARC policy applies to all of its subdomains. When retrieving DMARC policy records, if no record can be found at Header From FQDN level, receivers are obliged to determine the Organizational Domain [6]of the sender and lookup for a policy record there.

However, DMARC policy for an Organizational Domain can also specify a separate Subdomain Policy (“sp” tag of a DMARC record) that will apply for any subdomains that don’t have an explicit DMARC policy published.

In the scenario discussed earlier in the SPF chapter, you would:

1. Publish an explicit DMARC record for any subdomains that **are** legitimate sources of email.

2. Publish a Subdomain policy of “reject” in your Organizational Domain policy record to automatically reject any emails that spoof non-sending domains

This kind of structuring of your email authentication provides for the best possible protection of your infrastructure and brand.

DMARC-Specific Issues

There are several potential issues with DMARC, all of which come from the nature and shortcomings of other authentication technologies it relies to. The problem is that DMARC brought those issues to the surface by actively pushing a policy to reject the email, and by correlating all the different sender identifiers in a message.

Most issues occur with mailing lists and mailing list management software. When an email is sent to a mailing list, it is redistributed to all of its recipients. However, the resulting email, with a sender address of the original sender, will be delivered by the mailing list manager’s hosting infrastructure, thus failing SPF checks for Header From (most mailing list managers use the list address as Envelope From (MAIL FROM) and original sender’s address as Header From).

Since DMARC will fail for SPF, we may rely on DKIM, however, most mailing list managers also add footers to messages, or tag subjects with the list name, thus breaking DKIM signature verification.

Authors of DKIM suggest several solutions to the problem, all of which boil down to the mailing list managers having to use the address of the list in all From addresses, and indicating the original sender address by another means.

Similar problems arise from messages that are forwarded by just copying the original message over SMTP to the new recipient. However, most Mail User Agents in use today will correctly form a new message and include the forwarded message either inline or as an attachment to the new. Messages forwarded in this way will pass DMARC if the forwarding user passes (of course, the original message’s authenticity cannot be established).

Sample Action Plan To Implement Email Authentication

Although the technologies themselves are simple, the road to implement a complete email authentication infrastructure can be long and winding. For smaller organizations and those with controlled mail flows it will be fairly straightforward, while larger environments may find it exceptionally challenging. It is not uncommon for large enterprises to hire consulting help to manage the implementation project.,

Step 1: DKIM

DKIM is relatively unintrusive since unsigned messages will not incur any rejections. Before actual implementation, take into account all the points mentioned previously. Contact any third-parties that you might delegate signing to, make sure that your third-parties support DKIM signing, and consider your selector management strategy. Some organizations would keep separate keys (selectors) for different organizational units. You may consider the periodic rotation of keys for additional security – but make sure you don’t delete your old keys until all your messages in transit are delivered.

Special consideration should be taken to key sizes. Although in general “more is better”, you must take into account that creating two digital signatures per message (including canonicalization, etc.) is a very CPU-expensive task, and can influence the performance of outgoing mail gateways. Due to computation overhead, 2048 bits is the largest practical key size that can be used, but for most deployments, 1024-bit keys make a good compromise between performance and security.

For the successful subsequent implementation of DMARC, you should:

1. identify all domains that you send as, including subdomains
2. generate DKIM keys and create signing profiles for each domain
3. deliver relevant private keys to any third-parties
4. publish all public keys in relevant DNS zones
5. verify third-parties are ready to begin signing
6. turn on DKIM signing in RELAYED Mail Flow Policy on all your ESAs
7. notify third-parties to begin signing

Step 2: SPF

Properly implementing SPF will probably be the most time-consuming and cumbersome part of any email authentication infrastructure implementation. Because the email was very simple to use and manage, and completely open from security and access point of view, organizations historically didn’t enforce strict policies around who and how can use it. This resulted in most organizations today not having a complete view of all the different sources of email, both from within and externally. The single biggest problem of implementing SPF is to discover who is currently legitimately sending emails on your behalf.

Things to look for:

1. obvious targets – Exchange or other groupware servers or outgoing mail gateways
2. any DLP solutions or other email processing systems that may generate external notifications
3. CRM systems sending information interacting with customers
4. various third-party applications that may send email
5. lab, test or other servers that may send email
6. personal computers and devices configured to send an external email directly

The above list is not complete, as organizations have different environments, but should be considered as a general guideline as to what to look for. Once (most of) your email sources have been identified, you may want to take a step back, and instead of authorizing every single existing source, clean up the list. Ideally, all of your outgoing emails should be delivered through your outgoing mail gateways with a few justified exceptions. If you have your own or use a third-party marketing mail solution, you should use separate infrastructure than production email gateways. If your mail delivery network is exceptionally complicated, you may proceed with documenting the current state in your SPF, but do take time to clean the situation up in the future.

If you serve multiple domains over the same infrastructure, you may want to create a single universal SPF record and reference it in individual domains using the “include” mechanism. Make sure that your SPF records aren’t too wide; e.g. if only five machines in a /24 network send SMTP, add those five individual IP addresses to your SPF, rather than the entire network. Aim for your records to be as specific as possible to minimize any chances of malicious email compromising your identity.

Start off with a softfail option for non-matching senders (“~all”). Only change it to hardfail (-all) once you are 100% sure that you have identified **all** of your sources of email, otherwise you risk losing production email. Later on, after implementing DMARC and running it in monitor mode for a while, you will be able to identify any systems you missed and update your SPF records to be complete. Only then will it be safe to set your SPF to hardfail.

Step 3: DMARC

Once your DKIM and SPF are set up as complete as you can, it’s time to create your DMARC policies. Consider all the different situations mentioned in previous chapters, and prepare to deploy more than one DMARC record if you have a complex email infrastructure.

Create email aliases that will receive reports, or create a Web application that can ingest them. There are no strictly defined email addresses to be used for this, but it helps if they are descriptive, e.g. rua@domain.com, dmarc.rua@domain.com, mailauth-rua@domain.com, etc. Make sure you have a process in place for an operator to monitor these addresses and modify SPF, DKIM and DMARC configuration appropriately, or alert the security team in case of a spoofing campaign. Initially, the workload will be substantial as you tweak the records to cover anything you might have missed during SPF and DKIM configuration. After a while, reports will probably indicate only spoofing attempts.

Initially, set your DMARC policy to “none” and your forensic option to send reports for **any** failing checks (“fo=1”) – this will quickly discover any errors in your SPF and DKIM while not influencing traffic. Once you are happy with the contents of submitted reports, change the policy to “quarantine” or “reject”, depending on your security policy and preference. Again, make sure you have operators continuously analyzing your received DMARC reports for any false positives.

Implementing DMARC completely and correctly is not a small or short task. While some results (and formal “implementation” of DMARC) may be obtained by publishing an incomplete set of records and a policy of “none”, it is in the best interest of both the sender organization and the Internet as a whole that everyone implements it to the full extent of its capabilities.

Regarding timelines, here is a very rough outline of individual steps for a typical project. Again, as each organization is different, these are far from accurate:

1. DKIM planning and preparation	2-4 weeks
2. DKIM test runs	2 weeks
3. SPF – legitimate sender identification	2-4 weeks
4. DMARC policy preparation	2 weeks
5. SPF and DMARC records test run	4-8 weeks
6. SPF test run with hardfail	2 weeks
7. DMARC test run with quarantine/reject	4 weeks
8. Monitoring DMARC reports and adapting SPF/DKIM accordingly	continuous

Smaller organizations are likely to experience a shorter duration of most steps, especially Step 3 and 4. No matter how simple your email infrastructure you think may be, always allocate ample time during test runs, and monitor feedback reports closely for anything you might have missed.

Larger organizations might experience an even longer duration of the same steps, with more stringent test requirements. It is not uncommon for companies with complex email infrastructure to hire external help, not only for the technical aspect of email authentication implementation but also to manage the entire project and coordinate across teams and departments.

Additional References

- The reference site for SPF: http://www.openspf.org
- The DKIM Council: http://www.dkim.org
- DMARC main website, run by The Trusted Domain Project: http://www.dmarc.org
- dmarcian – a help and resources site run by Tim Draegen, one of the authors of DMARC. Make sure to visit the “Tools” section: http://www.dmarcian.com
- Online Trust Alliance’s Record Validator tool: https://otalliance.org/resources/spf-dmarc-record-validator
- DMARC Record Assistant – another useful tool to help you create your DMARC records: http://www.kitterman.com/dmarc/assistant.html
- SPF Record Testing Tools: http://www.kitterman.com/spf/validate.html
- “Don’t Be A Phish: Deep Dive Into Email Authentication Techniques”, a Cisco Live 2014 presentation BRKSEC-3770: https://www.ciscolive.com/online/connect/sessionDetail.ww?SESSION_ID=76627

[1] Canonicalization is beyond the scope of this document. Refer to material in “Additional References” section for further information on DKIM canonicalization.

[2] DKIM DNS record parameters are also out of scope of this document.

[3] Creation of Message Filters is beyond the scope of this document. Please refer to AsyncOS for Email user guides for assistance.

[4] M3AAWG defined an excellent set of best practices applied and honored by most of the industry. Their Sender Best Common Practices document is available at https://www.m3aawg.org/sites/maawg/files/news/M3AAWG_Senders_BCP_Ver3-2015-02.pdf

[5] This behavior takes advantage of the fact that originally, DKIM does not verify the message source as stated in MAIL FROM or Header From at all. It only verifies that the Signing Domain ID (“d” parameter of DKIM Signature, and “Domain Name” parameter in your Signing Profile) is indeed hosting the public key of the pair used to sign the message. Sender authenticity is implied by having the “From” header signed. Just make sure that you list any and all domains (and subdomains) you sign for in “Profile Users” section.

[6] Usually, a domain one level below TLD or relevant ccTLD prefix (.ac.uk, .com.sg etc…)