Communications Security Establishment
Centre de la sécurité des télécommunications

CANADIAN CENTRE FOR CYBER SECURITY

Implementation Guidance: Email Domain Protection

PRACTITIONER

Canada

FORWARD

ITSP.40.065 Implementation Guidance: Email Domain Protection is an UNCLASSIFIED publication that is issued under the authority of the Head of the Canadian Centre for Cyber Security (Cyber Centre). For more information, email or phone our contact centre:

Contact Centre
contact@cyber.gc.ca
(613) 949-7048 or 1-833-CYBER-88

EFFECTIVE DATE

This publication takes effect on August 12, 2021.

REVISION HISTORY

Revision	Amendments	Date
1	First release	April 7, 2020
1.1	Following changes made: - Added Annex C, Analyzing Email Information - Simplified the Enforce stage of the implementation plan - Removed mentions of the Cyber Centre DMARC reporting service - Clarified how DMARC handles domains, alignment, and the sp tag - Expanded guidance for subdomains and non-mail domains - Reorganized sections on related standards - Made various corrections and clarifications	August 12, 2021

ISBN 978-0-660-40018-1
CAT D97-3/40-065-2021E-PDF

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ANNEXES

1 OVERVIEW

This document provides guidance to system owners on implementing technical security measures to protect their domains from email spoofing. In this document, we describe technical measures that system owners can implement to prevent the delivery of certain malicious messages that are abusing the reputations of their domains and to identify the infrastructure used by malicious actors.

By implementing this guidance, you can prevent threat actors from successfully representing themselves as your organization by using your email domains. The guidance in this document also helps prevent phishing emails from being delivered to your organization.

Phishing is a widely used attack method in which threat actors send emails (or use other communication methods such as SMS or phone calls) that appear to come from a trusted source and that seem to be legitimate. Threat actors use phishing attacks to trick recipients into disclosing personal data and other sensitive information or as a method for installing malicious software (i.e. malware) on devices.

You can reduce a threat actor's chance of carrying out successful malicious email campaigns by implementing the technical security measures detailed in this document. These measures will protect your organization in the following ways:

- Preventing the delivery of malicious messages impersonating your domains;
- Deterring threat actors from attempting to spoof protected domains;
- Improving the security of email recipients; and
- Protecting the reputation of organizations whose domains are the target of spoofing.

2 EMAIL DOMAIN PROTECTION MECHANISMS

Three security protocols act jointly to protect email domains from being spoofed:

- Sender Policy Framework (SPF);
- DomainKeys Identified Mail (DKIM); and
- Domain-based Message Authentication, Reporting, and Conformance (DMARC).

For complete protection, you must implement all three protocols and configure them to instruct recipients to reject inauthentic messages. These protocols are described in the following sections. For guidance on implementing them, see Annex A.

2.1 SPF

You can use SPF to specify the Internet protocol (IP) addresses from which emails can be sent on a domain's behalf. The SPF standard is formally defined by the Internet Engineering Task Force's (IETF) *Request for Comments (RFC) 7208: Sender Policy Framework (SPF) for Authorizing Use of Domains in Email* [1][1].

You can implement SPF by publishing a domain name system (DNS) record for each of your domains and subdomains that lists the authorized IP addresses, either directly or by reference to other records.

When a message is received, an email system that supports SPF performs the following actions:

1. It retrieves the SPF record that is associated with the sending domain; and
2. It verifies that the IP address used to send the message has been authorized to do so.

Messages from unauthorized IP addresses may be accepted, marked as suspicious, or rejected; however, these actions depend on the policy that is included in the SPF record.

2.2 DKIM

DKIM provides a mechanism for email messages to be authenticated using a cryptographic signature. The DKIM standard is formally defined by IETF's *RFC 6376: DomainKeys Identified Mail (DKIM) Signatures* [2].

You can implement DKIM by:

1. Publishing, for each email domain, at least one DNS record consisting of a cryptographic public key and additional information;
2. Deploying the corresponding private keys to a domain's message transfer agents (MTAs); and
3. Configuring MTAs to sign outgoing messages.

You must configure each MTA with a private key that corresponds to a published DKIM record. When the MTA sends a message, it uses the private key to add a cryptographic signature to the message by inserting a message

[1] Numbers in square brackets refer to references cited in the Supporting Content section of this document.

header. If boilerplate content is added to outbound messages, for example a legal disclaimer, this must be done before the DKIM signature is applied. Otherwise, the signature will be invalidated.

When an email system that supports DKIM receives a DKIM-signed message, it retrieves the record associated with the message's DKIM header and verifies the message's signature using the published public key. This DKIM check cryptographically confirms that the message was sent by an authorized sender and was not altered in transit. If the signature is not valid, or if no DKIM record is available, the message will fail DKIM and may be rejected.

2.3 LIMITATIONS OF SPF AND DKIM

SPF and DKIM share a limitation that makes them ineffective against moderately sophisticated threat actors. Both protocols rely on domain names that are hidden from the user and that can differ from the domain that is displayed to the user in an email's `From` field (known as the *header from* address). SPF uses the domain of the SMTP HELO or *envelope from* email address (or alternatively the `Return-Path` address), which is used by email servers in the background. DKIM uses the domain that is specified in the DKIM header.

There is no requirement for the domains that are used in the *envelope from*, *header from*, or DKIM header to match. Therefore, a threat actor can increase the likelihood that malicious emails are delivered by implementing SPF and DKIM for a domain under their control, while presenting a different and more trustworthy domain in the message's `From` field to trick the recipient.

You should note that legitimate messages may be rejected by receiving systems if SPF or DKM records are missing or improperly configured. As these protocols do not include a reporting mechanism, you may not be aware of failed deliveries. Similarly, there is no mechanism that enables receiving systems to inform domain owners about messages that have been flagged as illegitimate and rejected due to a failed SPF or DKIM check.

2.4 DMARC

DMARC was created to address the limitations of SPF and DKIM and improve enforcement and reporting. The DMARC standard is formally defined by IETF's *RFC 7489: Domain-based Message Authentication, Reporting, and Conformance (DMARC)* [3].

DMARC introduces several improvements:

- Enforces alignment between the *envelope from* and *header from* domains for SPF;
- Enforces alignment between the DKIM header domain and the *header from* domain for DKIM;
- Allows system owners to specify the action that a receiving system should take if a message fails both the SPF and DKIM checks; and
- Provides a mechanism for receiving systems to report information on DMARC results to domain owners.

You can implement DMARC by publishing a DNS record for each of your domains that provides policy information to receiving systems.

2.4.1 DMARC VALIDATION

For an email to pass DMARC validation, it must pass either SPF or DKIM, and the domain used in those checks must align with the one in the email's `From` field. If an email fails both SPF and DKIM checks, including domain alignment, it fails DMARC. The mechanism by which a receiving system determines the DMARC policy to apply to a given message is detailed in Annex B, section B.3.

If an email passes DMARC validation, it is delivered. If, however, an email fails DMARC validation, the receiving email system applies the policy that is specified in the sending domain's DMARC record. The policy must be one of the following options:

- **None**: The email is delivered (i.e. monitor only);
- **Quarantine**: The email is delivered but is marked as suspicious; or
- **Reject**: The email is rejected.

A DMARC record can also specify that the published policy be applied to only a percentage of messages that fail validation, with the next less strict policy applied to the remainder. For example, a policy of **reject** with a percentage of 50% will reject 50% of messages which fail, while 50% will be quarantined.

While policies of **none** and **quarantine** can help you gather information and configure your systems, only a policy of **reject** at 100% will prevent all illegitimate messages from being delivered.

Figure 1 depicts the DMARC validation process.

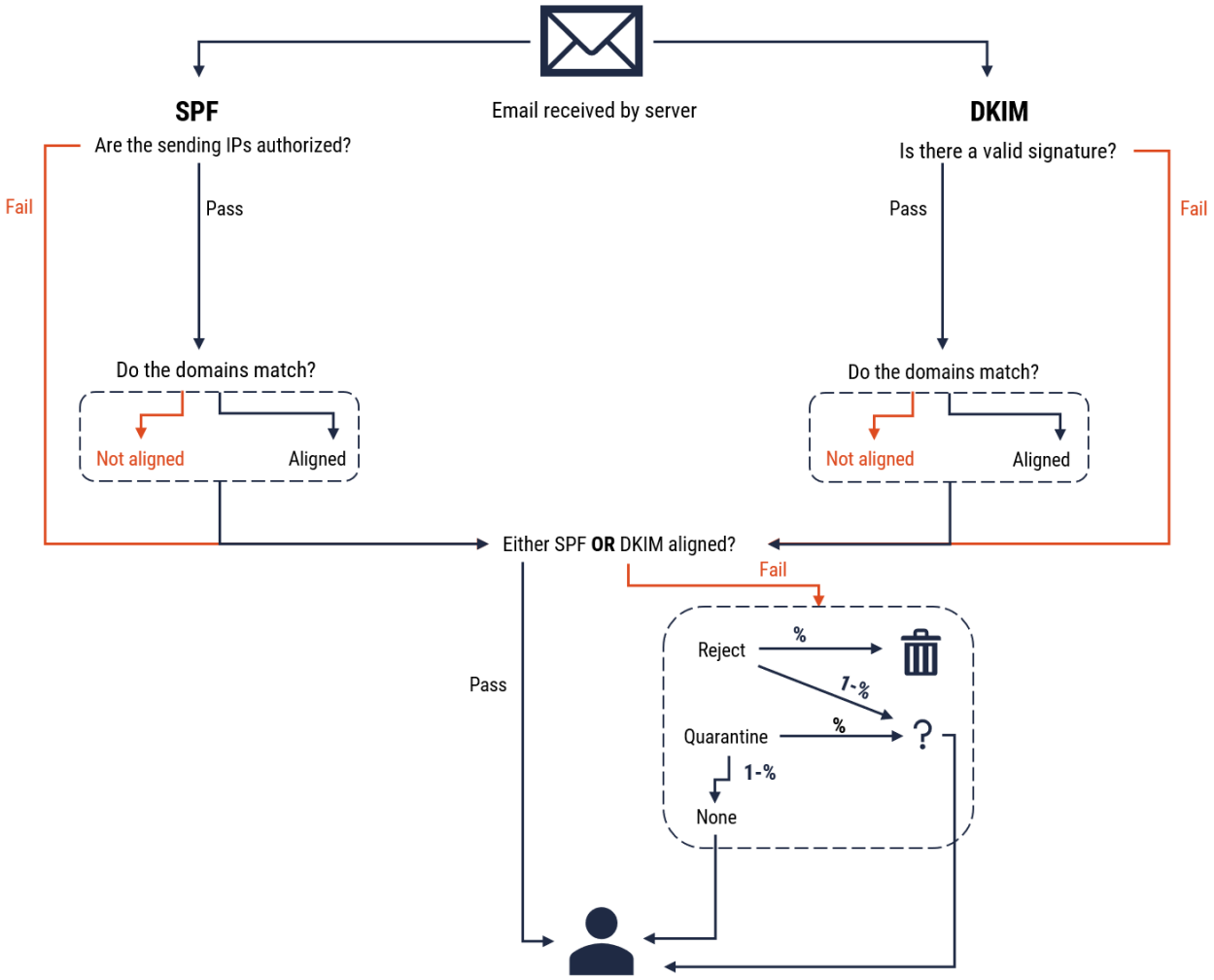

Figure 1: DMARC Validation

2.4.2 DMARC REPORTING

DMARC includes a mechanism for system owners to receive information about emails sent using one of their domains in the `From` field. You can use this information in the following ways:

- Identify the components of your organization's email infrastructure, including third-party senders;
- Confirm that SPF and DKIM have been deployed and are functioning correctly for your domains;
- Confirm that legitimate emails from your domains are being delivered successfully while illegitimate emails are rejected; and
- Identify infrastructure used by malicious actors to impersonate your domains.

DMARC aggregate reports are produced by receiving systems and are typically sent to domain owners daily. These reports are sent as email attachments to an address specified in a domain's DMARC record. While not all receiving email systems send aggregate reports, many large mail service providers do.

Aggregate reports are produced in a standardized Extensible Markup Language (XML) format and should be processed by an automated system wherever possible. There are many open source, free, and commercial offerings that can process DMARC reports on your behalf. Guidance on interpreting the information from DMARC aggregate reports is included in Annex C, section C.2.

Some receiving systems may also provide failure or forensic reports, which are copies of messages that have failed one or more DMARC checks. Systems owners can request forensic reports by specifying certain parameters in a domain's DMARC record. However, we do not recommend that system owners request forensic reports, as they may potentially contain personally identifiable information (PII).

3 ADDITIONAL CONSIDERATIONS

3.1 VENDOR SUPPORT

Not all hardware, software, or service providers fully implement support for SPF, DKIM, and DMARC. Check the documentation for the components and services used in your infrastructure to confirm the level of support and identify any potential limitations.

Receiving systems may not fully implement the checks for SPF, DKIM, or DMARC, or they may ignore or override an organization's published policy. For example, most receivers who send DMARC aggregate reports do so once every 24 hours, regardless of the reporting period published in the policy for the sending domain.

3.2 THIRD-PARTY SENDERS

When implementing SPF, DKIM, and DMARC, you should consider the third parties that have been authorized to send emails on behalf of a domain.

3.2.1 THIRD PARTIES AND SPF

For emails sent by third parties to pass SPF, the sending IP addresses must be incorporated into a domain's SPF record. To accomplish this, the domain's SPF record can list the IP addresses directly or include a reference to an SPF record that is maintained by the third party.

Note that the SPF protocol enforces a limit of 10 DNS lookups per record, which can inadvertently be exceeded depending on how referenced third-party records have been structured.

3.2.2 THIRD PARTIES AND DKIM

For emails sent by third parties to pass DKIM, they must be signed using a private key with a corresponding DKIM record published for the domain. Third parties typically accomplish this by requesting system owners to publish either a provided DKIM record or a CNAME record that redirects to a DKIM record maintained by the third party. In the latter case, the third party typically manages and rotates the cryptographic keys used, though, as a system owner, you may also be able to request that the keys be rotated on demand.

3.2.3 THIRD PARTIES AND DMARC

As noted above, DMARC enforces alignment between the *envelope from* and *header from* domains for SPF and DKIM to pass. In some cases, this causes complications if third-party senders are used, and the two domains do not match.

You can ensure DKIM alignment by using a CNAME DKIM record, as described above. To address SPF misalignment, you can create a designated subdomain to be used as part of a custom `Return-Path` address, along with a corresponding CNAME record that redirects to an SPF record maintained by the third party.

You can also direct DMARC reports to a third party for processing. You can implement this by including a third party's email address as a reporting destination in a domain's DMARC record. In such a case, the third party must also publish a corresponding record indicating that their domain is willing to accept DMARC reports on behalf of your domain.

3.2.4 SUBDOMAIN SEPARATION

To protect user addresses from being spoofed, you should consider allocating dedicated subdomains to authorized third-party senders. For example, a third party could be authorized to send messages from `list.domain.example`, but not from `domain.example`. As a result, the third party would not be able to send an authorized message from a user address such as `chief.executive@domain.example`, even in the case of a compromise in their system.

3.3 MESSAGE FORWARDING

Forwarding messages can sometimes cause SPF, DKIM, or DMARC checks to fail, depending on how the message is handled. Typically, a user can forward a message using their email client application without issue. However, there are sometimes issues when email systems forward messages automatically. For example, if the *envelope from* address is changed but the *header from* address is not, SPF validation or alignment can fail. Similarly, if any of the content underlying the signature in a DKIM header is changed, the signature validation will fail.

You cannot prevent messages from being forwarded in this way, but you should be aware that automatically forwarded messages may be rejected by receiving systems in these cases. DMARC failures are noted in the aggregate reports you receive; review these reports to assess any potential problems.

Overall, DKIM is more resilient to forwarding than SPF. Ensuring that DKIM is properly deployed helps to ensure the delivery of forwarded legitimate messages.

3.3.1 AUTHENTICATED RECEIVED CHAIN (ARC)

ARC is a draft standard that attempts to address the problems caused by forwarded messages. The ARC protocol is described in *RFC 8617: The Authenticated Received Chain (ARC) Protocol* [4], which was granted experimental status in July 2019. ARC allows intermediate systems to validate the headers of an email with respect to SPF and DKIM and attest to their validity with a digital signature when forwarding the message. Downstream systems can rely on this chain of attestations and choose to deliver a message, even if it has failed the receiving system's own DMARC checks.

ARC is an emerging standard. However, some of the Internet's largest email providers have adopted the signing and validation of messages using ARC. As a result, you may notice that ARC is mentioned in a provider's DMARC reports, particularly when a domain's DMARC policy has been overridden because of a successful ARC validation. More information on ARC is provided in Annex C, section C.2.

3.4 INBOUND MAIL

You should configure your organization's mail infrastructure to support authentication mechanisms for received mail, including the following:

- Attempt to validate SPF, DKIM, and DMARC for all received messages;
- Reject messages according to a domain's SPF and DMARC policies;
- Preserve message content and DKIM headers when forwarding messages; and
- Consider sending aggregate DMARC reports to domain owners regarding messages received.

3.5 EMAIL TRANSPORT ENCRYPTION

Historically, network traffic between mail servers was unencrypted, leaving it vulnerable to interception or modification in transit. Over time, several protocols have emerged to support the encryption of email traffic in transit.

3.5.1 STARTTLS

In 2002, the Simple Mail Transfer Protocol (SMTP) was extended to support opportunistic encryption via a STARTTLS command. You should configure your organization's MTAs to support and request the use of STARTTLS. However, an attacker in a privileged network position can prevent the delivery of this command, potentially limiting the effectiveness of this approach in some cases.

3.5.2 DNS-BASED AUTHENTICATION OF NAMED ENTITIES (DANE)

DANE was introduced in 2012 to allow system owners to bind transport layer security (TLS) certificates to domain names, without relying on a central certificate authority. To address the limitations of STARTTLS, *RFC 7672: SMTP Security via Opportunistic DANE TLS* [5] describes the application of DANE to SMTP traffic, which allows system owners to require that traffic to their domain's mail servers be encrypted.

To deploy DANE, you must also adopt the prerequisite of Domain Name System Security Extensions (DNSSEC) for a domain. System owners with a DNSSEC-enabled domain can implement DANE by publishing a DANE TLS authentication (TLSA) record in DNS. Like DMARC, DANE supports the delivery of aggregate reporting to system owners.

3.5.3 MTA STRICT TRANSPORT SECURITY (MTA-STS)

As an alternative to DANE, SMTP MTA-STS is an emerging standard that adds support for strict encryption without relying on DNSSEC. With MTA-STS, you can specify that mail traffic sent to a domain is encrypted with a specific public encryption key.

While they achieve the same aim, DANE and MTA-STS do not conflict and you can implement them in parallel. You can implement MTA-STS by deploying signed TLS certificates to a domain's email and web servers, publishing an

MTA-STS policy on the web server, and publishing MTA-STS records in DNS. Like DMARC, MTA-STS supports the delivery of aggregate reporting to system owners.

3.6 NON-MAIL DOMAINS

Domains that are completely unused or not used to send email should also be protected against spoofing by creating the requisite DNS records. DNS record guidance specific to non-mail domains is included at Annex B, section B.4.

3.7 BRAND INDICATORS FOR MESSAGE IDENTIFICATION (BIMI)

BIMI is an emerging standard that will allow system owners to provide an image, typically an organization's logo, which email clients will display to users alongside authenticated messages. To participate in BIMI, a domain must have fully implemented DMARC and have a policy of **reject**. BIMI is not yet widely adopted, but its development is supported by many large stakeholders in the email community.

4 SUMMARY

You can use the guidance in this document to implement technical security measures to protect your organization's domains from email spoofing. By implementing SPF, DKIM, and DMARC, you can reduce a threat actor's chance of carrying out successful malicious email campaigns leveraging the reputation of your organization.

Annex A of this document provides guidance on how to implement these three security protocols, Annex B includes a protocol reference, and Annex C provides guidance on how to analyze email information.

4.1 CONTACT INFORMATION

For more information on implementing protection measures for email domains, email or phone our Contact Centre:

Cyber Centre Contact Centre

contact@cyber.gc.ca

(613) 949-7048 or 1-833-CYBER-88

5 SUPPORTING CONTENT

5.1 LIST OF ABBREVIATIONS

Term	Definition
A	DNS host record (IPv4)
ARC	Authenticated Received Chain
BIMI	Brand Indicators for Message Identification
CSE	Communications Security Establishment
DANE	DNS-based Authentication of Named Entities
DKIM	DomainKeys Identified Mail
DMARC	Domain-based Message Authentication, Reporting, and Conformance
DNS	Domain Name System
DNSSEC	Domain Name System Security Extensions
GC	Government of Canada
IETF	Internet Engineering Task Force
IP	Internet Protocol
MTA	Mail Transfer Agent
MTA-STS	SMTP MTA Strict Transport Security
MX	DNS mail exchange record
PII	Personally Identifiable Information
RFC	Request for Comments
SMTP	Simple Mail Transfer Protocol
SPF	Sender Policy Framework
TLD	Top-level domain
TLS	Transport Layer Security
TLSA	Transport Layer Security Authentication
TTL	Time to Live
TXT	DNS text record
XML	Extensible Markup Language

5.2 REFERENCES

Number	Reference
1	Internet Engineering Task Force. *RFC 7208 Sender Policy Framework (SPF) for Authorizing Use of Domains in Email, Version 1*. April 2014.
2	Internet Engineering Task Force. *RFC 6376 DomainKeys Identified Mail (DKIM) Signatures*. September 2011.
3	Internet Engineering Task Force. *RFC 7489 Domain-based Message Authentication, Reporting, and Conformance (DMARC)*. March 2015.
4	Internet Engineering Task Force. *RFC 8617 The Authenticated Received Chain (ARC) Protocol*. July 2019.
5	Internet Engineering Task Force. *RFC 7672 SMTP Security via Opportunistic Domain-Based Authentication of Named Entities (DANE) Transport Layer Security (TLS).* October 2015.
6	Mozilla Foundation. *Public Suffix List*. https://publicsuffix.org/
7	Internet Engineering Task Force. *RFC 8601 Message Header Field for Indicating Message Authentication Status*. May 2019.
8	National Institute of Standards and Technology. SP 800-177 Rev. 1 *Trustworthy Email*. February 2019.

Annex A Implementation Plan

A.1 Overview

To implement SPF, DKIM, and DMARC, you need to take several incremental steps. To minimize potential disruptions, we recommend that you follow the sequence below. Note that each step is described further in the following subsections of this annex.

1. **Assess:**
    a. Identify all domains and subdomains used to send mail;
    b. Assess current state;
    c. Deploy initial DMARC records with policy of **none**; and
    d. Collect and analyze DMARC reports.
2. **Deploy:**
    a. Identify all authorized senders;
    b. Deploy SPF records for all domains;
    c. Deploy DKIM records and keys for all domains and senders; and
    d. Monitor DMARC reports and correct misconfigurations.
3. **Enforce:**
    a. Upgrade DMARC policy to **quarantine**;
    b. Upgrade DMARC policy to **reject**; and
    c. Reject all messages from non-mail domains.
4. **Maintain:**
    a. Monitor DMARC reports;
    b. Correct misconfigurations and update records as required; and
    c. Rotate DKIM keys annually.

A.2 Assess

A.2.1 Identify Mail Domains

Compile a list of all domains and subdomains from which your organization sends email. You can acquire this information from some of the following sources:

- Existing DNS records of type MX;
- Logs of network traffic on ports 25 or 587 (typically used for SMTP); and
- Email administrators.

A.2.2 Assess Current State

Once you have identified the domains in use, use DNS tools to assess the current state of SPF, DKIM, and DMARC deployment for each.

Note that you cannot test DKIM records in this manner unless you also know the corresponding selector string, which you can find in the `DKIM-Signature` header of a message from the domain (see Annex C, section C.1 for more information on email headers). Some email service providers also use well-known selectors for all their customers.

Review returned records to ensure they are correct (refer to Annex B for more information on the required structure and content). There are many online tools that will assess the state of a given domain name, including validating the correctness of the returned records.

You can use the following `dig` commands to retrieve existing records for each protocol:

- **SPF:** `dig +short -t txt domain.example`
- **DKIM:** `dig +short -t txt selector._domainkey.domain.example`
- **DMARC:** `dig +short -t txt _dmarc.domain.example`

A.2.3 Deploy Initial DMARC record

Deploy an initial DMARC record to each domain and subdomain that is used to send email. By deploying an initial DMARC record with a policy of **none**, you can immediately begin to receive DMARC reports, even before deploying SPF or DKIM. The initial record does not impact mail delivery in any way, but it requests that receiving systems send aggregate DMARC reports to you and/or a designated third party. You can use these reports to guide your SPF and DKIM deployments, and they can identify mail infrastructure that may not have been known at the outset.

Your organization must specify an email address to which the reports are sent. This address should be a dedicated mailbox (e.g. `dmarc@domain.example`) rather than an address assigned to a named individual.

We recommend the following form for your organization's initial DMARC record:

```
v=DMARC1; p=none; sp=none; rua=mailto:dmarc@domain.example
```

You can specify a third party's email address in the record to have reports sent to that third party. To receive these reports, the third-party domain must also publish a DMARC record indicating that they accept reports on behalf of the domain, which is the subject of the report.

You can specify up to two mailboxes; however, some senders may only send reports to the first address listed. An example record for this case is as follows:

```
v=DMARC1; p=none; sp=none;
rua=mailto:dmarc@thirdparty.example,mailto:dmarc@domain.example
```

A.2.4 Collect and Analyze DMARC Reports

Within 24 hours of publishing an initial DMARC record, you should begin to receive aggregate reports from systems that have received emails from your organization's domain. Monitor the DMARC reports received throughout the deployment process, as they contain valuable information that you can use to support your efforts.

DMARC reports are generated in a standardized XML format and should be processed by an automated system. There are many open source, free, and commercial options available for DMARC report processing.

When analyzing DMARC reports during this stage, you should pay attention to the following components:

- **Subdomains**: Identify subdomains that are used to send email and verify whether these subdomains are configured with SPF, DKIM, or DMARC.
- **Internal IP addresses**: Identify internal IP addresses that are used to send email and verify whether the sent messages pass SPF and DKIM alignment as expected.
- **External IP addresses**: Identify any external IP addresses that are used to send email and use reverse DNS and WHOIS tools to gather more information about them. Attempt to classify these addresses as one of the following:
  - **Authorized senders**: Third parties who are authorized to send email on your organization's behalf. Verify whether the sent messages pass SPF and DKIM alignment as expected;
  - **Forwarders / Relays**: Some mail servers, such as those supporting mailing lists, may be configured to automatically forward messages. These forwarded messages may fail SPF and DKIM checks when received but are benign; or
  - **Possible spoofing**: IP addresses that do not fall into one of the above categories may be sending unauthorized messages that are spoofing your organization's domain. IP addresses that appear on public spam blacklists are more likely to fall into this category.

Additional guidance on analyzing DMARC reports is included in Annex C.

A.3 Deploy

A.3.1 Identify Authorized Senders

By analyzing the DMARC reports that you receive you can identify all authorized senders, including subdomains and IP addresses. This list likely includes internal infrastructure, but it can also include cloud-based email services or third-party senders.

You may need to consult other business owners in your organization to confirm whether certain senders (e.g. a third-party firm contracted to send marketing emails) are authorized.

A.3.2 Configure DNS Time To Live (TTL)

DNS records are cached for a period known as their time to live (TTL), expressed in seconds. During the Deploy and Enforce phases, you should use a TTL of 30 minutes (1800 seconds) for newly created or amended records. This will prevent any inadvertent errors from persisting in DNS caches for a longer period before they are replaced by a corrected version.

When revising a record, you should first modify the TTL of the existing record to 30 minutes and allow the previous TTL to elapse. This will ensure that the new record will replace the old one in all caches within 30 minutes.

You should use a TTL of 24 hours (86400 seconds) for records that are stable.

A.3.3 Deploy SPF for All Domains

For each domain and subdomain that sends email, deploy SPF records in the following sequence:

1. Construct an SPF record as described in Annex B.
   Note: The record must list all authorized IP addresses, either directly or by reference.
2. Publish the record in DNS.

A.3.4 Deploy DKIM for All Domains and Senders

For each domain that sends email, deploy DKIM keys and records in the following sequence:

1. Generate or acquire cryptographic keys for use with DKIM through one of the following methods:
   a. MTAs may have the ability to generate DKIM key pairs directly and output the public key.
   b. You can generate key pairs using a tool such as OpenSSL as follows:

   ```
   openssl genrsa -out private2048.key 2048
   openssl rsa -in private2048.key -pubout -out public2048.key
   ```

   c. Third-party senders may manage DKIM keys internally and provide you with a public key.

   Note: Key pairs should meet the cryptographic requirements specified in Annex B.
2. Use the public key(s) to construct DKIM records according to Annex B.
3. Determine strings to use as DKIM selectors or obtain specified selectors from third parties.

4. Publish DKIM records in DNS.
5. Take the following actions for each MTA:
    a. Configure a private key and the corresponding selector to use for DKIM.
    b. Activate DKIM signing of outgoing messages.

A.3.5 Monitor DMARC Reports and Correct Misconfigurations

Once you have deployed SPF and DKIM records, monitor the DMARC reports you receive to detect any errors or oversights. In particular, examine the reports for any instances of messages sent from authorized senders that fail either SPF or DKIM alignment. Correct any misconfigurations.

A.4 Enforce

When you are confident that your organization's SPF and DKIM deployments are complete and all misconfigurations have been corrected, you can transition the domains to enforcement. This change advises receiving mail systems to only deliver messages that pass DMARC (i.e. pass either SPF or DKIM alignment) and either quarantine or reject messages that fail DMARC (i.e. fail both SPF and DKIM alignment).

An enforcement policy must be specified for a domain using the DMARC record's `p` tag and must be one of the following three levels of increasingly progressive strictness:

- **None**: All messages are delivered (i.e. monitor only);
- **Quarantine**: Messages that fail DMARC are delivered but are marked as suspicious; or
- **Reject**: Messages that fail DMARC are rejected.

In the *disposition* field, DMARC reports will indicate the enforcement policy that has been applied to a specific group of messages.

A.4.1 Gradually Increase Enforcement

You should gradually increase the strictness of enforcement for a domain from a baseline of **none**, through **quarantine**, to an end state of **reject** as follows:

1. Use the following sequence of DMARC records:
    a. `v=DMARC1; p=none; rua=mailto:dmarc@domain.example`
    b. `v=DMARC1; p=quarantine; pct=100; rua=mailto:dmarc@domain.example`
    c. `v=DMARC1; p=reject; pct=100; rua=mailto:dmarc@domain.example`
2. At each stage, monitor DMARC reports to ensure that legitimate messages are not quarantined.
3. Once any detected errors have been corrected and confirmed by subsequent reports, progress to the next value.

In addition to the level of enforcement, a DMARC policy can specify a percentage of failing messages to which the enforcement level is applied, with the next lower level applied to the remainder. This is done using the DMARC record's optional `pct` tag, and you can use this adjustment to add more fine-grained increments to the progression above if desired.

For example, a policy of **quarantine** with a percentage of 50% would instruct a receiving system to quarantine 50% of failed messages, and to deliver the remaining 50%. The corresponding DMARC record would be as follows:

```
v=DMARC1; p=quarantine; pct=50; rua=mailto:dmarc@domain.example
```

A.4.2 Subdomain Exceptions

If subdomains of the root domain are used to send email and require a different configuration than the organizational domain, they can be configured with a unique DMARC record. This record will take precedence over that of the organizational domain. This can be useful in situations where a specific subdomain may take longer to bring into SPF and DKIM compliance than the organizational domain, allowing it to temporarily remain at a lower level of enforcement while that of the organizational domain (and other subdomains) is increased.

For subdomains without their own DMARC record, the policy of the organizational domain will apply. In particular, the policy specified by the `sp` tag will be used if it is present; otherwise, the policy of the `p` tag will be used.

Annex B, section B.3 details some specific considerations with respect to DMARC records and subdomains.

A.4.3 Non-Mail Domains

Once DMARC is deployed for all mail domains, you should also deploy SPF and DMARC records to protect domains that do not send mail. These records indicate that any messages received from these domains are illegitimate and should be rejected. Recommended records for these domains are included in Annex B, section B.4.

A.5 Maintain

A.5.1 Monitor DMARC Reports

Once SPF, DKIM, and DMARC are deployed, you should monitor DMARC reports for any changes. Pay attention to the following components:

- An increase in the number of messages failing SPF or DKIM alignment;
- The use of new subdomains;
- The use of new internal IP addresses;
- The ongoing use of new external IP addresses, which may indicate new third-party senders; and
- The volume of suspected spoofing.

A.5.2 Correct Misconfigurations and Update Records

As infrastructure changes and misconfigurations are uncovered, you will need to periodically adjust the SPF, DKIM, and DMARC records. You may also need to deploy new records in response to changes in your organization's domain structure.

SPF records are tied to IP addresses and are likely to require the most attention. Using CNAME records or `include` tags for third-party senders and `a` or `mx` tags for internal infrastructure can help to insulate SPF records from these kinds of changes, as they are not tied to IP addresses directly. Similarly, you may need to add or change DKIM records as you add or remove authorized senders or MTAs.

Once your organization has reached a policy of **reject**, DMARC records should remain largely stable.

A.5.3 Rotate DKIM Keys

To guard against the potential compromise of one or more private keys, you should rotate DKIM key pairs annually. You should rotate key pairs that you manage according to the following sequence:

1. Generate new key pairs.
2. Choose or obtain new selectors.
3. Prepare and publish new records with the public keys.
4. Configure MTAs to use the new private keys and selectors.
5. Delete the old records after one week.

Confirm the key rotation policy of third-party senders and ensure that their keys are rotated at least annually.

Annex B Protocol Reference

B.1 SPF

B.1.1 SPF Records

SPF records are published as TXT records at the root of a sending domain and at each subdomain. Each record includes a list of all the IP addresses that are permitted to send email on behalf of the domain, either directly or by referring to further DNS records that contain these addresses.

Stable SPF records should have a long time -to -live (TTL), such as 24 hours (86400 seconds).

Elements in SPF records must be separated by spaces and are assumed to be positive or permissive if they are not preceded by a modifier character (e.g. –) to indicate otherwise. A typical SPF record for the domain `domain.example` would be:

```
v=spf1 mx a:mail.domain.example include:spf.third-party.example -all
```

Table 1 describes the most commonly used elements of an SPF record.

Table 1: SPF Record Elements

Element	Description
`v=spf1`	The mandatory header for an SPF record.
`mx`	Incorporates any MX (mail) records published for the domain. Can be scoped to a specified domain or hostname using the form "`a:host1.domain.example`".
`a`	Incorporates any A (IPv4 address) or AAAA (IPv6 address) records published for the domain. Can be scoped to a specified domain or hostname using the form "`a:host1.domain.example`".
`ip4`	An IPv4 address or range, e.g. `ip4:192.0.2.0/24`
`ip6`	An IPv6 address or range, e.g. `ip6:2001:DB8::/32`
`include`	Incorporates SPF records published at another DNS location. Typically used to authorize third-party senders.
`-all`	Excludes all other IP addresses. Should be used as the last element of a record to indicate that all other senders are unauthorized.

B.1.2 Third-Party Senders

Third-party infrastructure is typically authorized using an `include` clause, which instructs the receiving email server to include any SPF records found at the specified location. For example:

```
include:spf.third-party.example
```

B.1.3 DNS Lookup Limit

The SPF protocol has a limit of 10 DNS lookups per record. If multiple third-party `include` statements are used, this limit can inadvertently be exceeded, especially if the included records also have `include` clauses of their own.

One way to work around this limit is to separate third-party senders into distinct subdomains. As a result, the SPF records for each subdomain would require fewer SPF elements and associated lookups than would be needed in a combined record.

B.2 DKIM

B.2.1 DKIM Records

DKIM records are TXT records that are specified by a selector string. They are created below a `_domainkey` location, situated one level below the associated domain. For example, the location of a DKIM record for the domain `domain.example` and with the selector `abc` is `abc._domainkey.domain.example`.
We recommend incorporating the date when a key pair was generated into the selector as a reminder of the key rotation period.

Stable DKIM records should have a long time to live (TTL), such as 24 hours (86400 seconds).

Elements in DKIM records must be separated by semicolons. Table 2 describes the typical elements of a DKIM record.

Table 2: DKIM Record Elements

Element	Description
`v=DKIM1`	The mandatory header for a DKIM record.
`p`	Public key value

B.2.2 Cryptographic Considerations

You should configure DKIM to use the RSA-SHA256 algorithm, and you should use keys that are 2048 bits in length. If an appliance or service cannot support 2048-bit keys, you should use 1024-bit keys.

Note that because 2048-bit keys exceed the limit of 255 characters for a single TXT record, they must span two adjacent records without additional spaces inserted. You should also take care to remove any non-visible new-line characters from the end of record strings.

To reduce the potential risk from a compromised private key, use unique DKIM key pairs as follows:

- Use a unique key pair, at a minimum, for each domain;
- Use unique key pairs whenever a third party manages the private key; and
- Use a unique key pair for each MTA, if feasible.

You should rotate DKIM keys on an annual basis as described in Annex A.

B.2.3 Third-Party Senders

Third-party infrastructure is typically authorized by using a specific selector associated with the key pair used by the sender. In some cases, the third party may manage the keys and provide you with the public key and selector to be used in a corresponding DKIM DNS record. Alternatively, the third party may provide you with a DNS location that you publish as a CNAME record in your domain and that points to a DKIM record maintained by the third party.

Some third-party senders will rotate DKIM keys managed by them automatically, while others require an intervention from the system owner to initiate the rotation.

B.3 DMARC

B.3.1 DMARC Records

DMARC records are TXT records created at the `_dmarc` location below the associated domain (i.e. `_dmarc.domain.example`).

Stable DMARC records should have a long time to live (TTL), such as 24 hours (86400 seconds).

DMARC record elements must be separated by semicolons. Table 3 lists the typical elements of a DMARC record.

Table 3: DMARC Record Elements

Element	Required	Description
`v=DMARC1`	Required	The mandatory header for a DMARC record.
`p`	Required	The policy to apply to messages that fail DMARC. Must be either “none”, “quarantine”, or “reject”.
`aspf`	Optional	Specifies whether the matching of domain names for SPF alignment should be relaxed or strict. Relaxed alignment is the default and is recommended in most cases.
`adkim`	Optional	Specifies whether the matching of domain names for DKIM alignment should be relaxed or strict. Relaxed alignment is the default and is recommended in most cases.
`sp`	Optional	The policy to apply to subdomains that do not have their own DMARC record. Must be either “none”, “quarantine”, or “reject”. This tag is only applicable to the records of organizational domains (i.e. domains that are one level below a domain published on the Public Suffix List [6]) and will be ignored at other levels of the domain hierarchy. If not present, the value of the `p` tag will be used for subdomains.
`pct`	Optional	The percentage of failed messages to which a “quarantine” or “reject” decision should be applied, with the next lower policy applied to the next lower level, i.e. “none” instead of “quarantine” or “quarantine” instead of “reject”. Must be an integer between 1 and 100; defaults to 100 if not present.
`rua`	Optional	The address(es) to which aggregate reports should be sent. Up to two addresses can be specified, separated by commas and with each preceded by `mailto:`. This use of this tag is strongly encouraged.
`ruf`	Optional	The address(es) to which forensic or failure reports should be sent. The use of this tag is discouraged due to the potential of collecting private communications.

We recommend the following initial DMARC record (with your own report reception mailbox):

```
v=DMARC1; p=none; sp=none; rua=mailto:dmarc@domain.example
```

When implementation is complete, the record should resemble the following:

```
v=DMARC1; p=reject; pct=100; sp=reject;
rua=mailto:dmarc@domain.example
```

B.3.2 DMARC Policy Selection

When determining which DMARC policy to apply, a receiving system will search for a DMARC record associated with the domain of the message's `From` header. DMARC uses the concept of an "organizational domain", which is tied to a list of domain suffixes known as the Public Suffix List [6]. This list includes top-level domains (TLDs) such `com`, `org`, and `ca`, as well as some lower-level suffixes such as `gc.ca`. An organizational domain is any domain that is one level below a suffix published on the list, e.g. `canada.ca` or `cyber.gc.ca`.

For a given message, the receiving system will use the first valid DMARC record that is found according to the following sequence:

1. A record that exactly matches the `From` header's domain.
2. If the `From` header's domain is a subdomain of an organizational domain:
    a. The record of the organizational domain will be used. This applies to subdomains at any level, skipping any intermediate subdomain levels.
    b. If a subdomain policy (`sp`) tag is present in the organizational domain's record, the policy specified by this tag will used. Otherwise, the policy of the `p` tag will be used.

For example, for a message with a `From` domain of `sub2.sub1.domain.example`, the receiver will apply the first policy that it finds from the following sequence:

1. `_dmarc.sub2.sub1.domain.example` → `p` tag.
2. `_dmarc.domain.example` → `sp` tag.
3. `_dmarc.domain.example` → `p` tag.

B.3.3 DMARC Domain Alignment

For the purposes of DMARC, SPF and DKIM domains are considered to be aligned with the `From` header's domain if they match according to the respective strictness levels specified in the DMARC record. There are two levels of strictness, which can be configured separately for SPF and DKIM:

- **Relaxed (default)**: Domains are aligned if they have the same organizational domain; and
- **Strict**: Domains are aligned only if they are an exact match. To meet strict alignment, additional SPF, DKIM, and DMARC records for subdomains may be required.

B.4 Non-Mail Domains

B.4.1 No Service MX Record

While SPF can be used to authorize outbound senders, it does not state if a domain is meant to receive email. No Service or Null MX responses were proposed in 2015 as a method to publish the lack of expected infrastructure to receive messages. Configuring a Null MX response is an improved method to advertise when a domain is not expected to receive messages.

The recommended MX record for non-mail domains is:

Preference: `0` (zero)
Hostname: `.` (period)

This must be the only MX record for the domain.

B.4.2 SPF

SPF records for non-mail domains should be configured to reject messages sent from any IP address.
The recommended SPF record for non-mail domains is:

```
v=spf1 -all
```

This record should be created as a wildcard TXT entry so that it will be returned in response to TXT queries for the root domain or any subdomains.

B.4.3 DMARC

DMARC records for non-mail domains should be configured to reject all messages that fail SPF and DKIM, which will be all messages when combined with the SPF record above. The recommended DMARC record for non-mail domains is:

```
v=DMARC1; p=reject; sp=reject; pct=100; rua=mailto:dmarc@domain.example
```

This record does not need to be created as a wildcard; the DMARC protocol specifies that the record at the organizational domain level should be used if one for a specific subdomain is not found.

Annex C Analyzing Email Information

When implementing SPF, DKIM, and DMARC, system administrators can use the information provided in email headers and DMARC aggregate reports to diagnose misconfigurations and troubleshoot delivery issues.

C.1 Email Headers

While not typically visible to end users, email headers contain a wealth of information that can be useful to system administrators. Headers are added to messages at various stages, usually by systems that have interacted with the message in some way. Headers can typically be accessed in an email client via an option to view the "original" message or its properties.

Table 4 describes the most useful headers in the context of email domain protection.

Table 4: Email Headers

Header	Description
`ARC-Authentication-Results`	The results of authentication checks performed by a system that has forwarded a message.
`ARC-Message-Signature`	A signature applied by a forwarding system to attest to the contents of the original message headers.
`ARC-Seal`	A signature generated by a forwarding system to attest to the integrity of the `ARC-Authentication-Results` and `ARC-Message-Signature` headers.
`Authentication-Results`	Records the results of authentication checks performed by receiving system, including SPF, DKIM, and DMARC. See section C.1.1.
`Delivered-To`	The mailbox to which a message was ultimately delivered.
`DKIM-Signature`	The DKIM signature applied to a message by the originating system. See section C.1.2.
`From`	The display name and the source address that most email clients present to the user. The domain from this address is used for DMARC authentication.
`Received`	A set of headers added to by each MTA that handles a message from its origin to its destination. These are usually prepended and can be read in reverse chronological order. Each header typically contains the hostnames of the sending and receiving hosts and a timestamp and can also include the associated IP addresses and connection information.
`Received-SPF`	Records the results of SPF verification by a receiving system. See section C.1.3.
`Return-Path`	Specifies an email address to which error messages should be sent. When present, the domain from this header will be used to evaluate SPF

	instead of the domain from the sending server's SMTP HELO message. When using third-party senders, specifying a `Return-Path` address that matches the `From` header can allow messages to pass SPF alignment.
`To`	The destination address of the message.
`x-[system-specific]`	Any headers beginning with "`x-`" are non-standard. These are typically system-specific headers that a vendor or service provider has added for their own use. In some cases, they may provide additional information on how a message was handled, for example how it was evaluated by a security or anti-spam service.

C.1.1 Authentication-Results

The Authentication-Results header summarizes the results of SPF, DKIM, and DMARC authentication. Described in *RFC 8601 Message Header Field for Indicating Message Authentication Status* [7], this header:

> "…consists of an authentication identifier, an optional version, and then a series of statements and supporting data. The statements are of the form "method=result" and indicate which authentication method(s) were applied and their respective results. For each such statement, the supporting data can include a "reason" string [in parentheses] and one or more "property=value" statements indicating which message properties were evaluated to reach that conclusion."

The identifier and authentication statements are separated by semicolons and can be more easily deciphered when split into separate lines as in the following example:

```
Authentication-Results:
mta2.receiver.example;
dkim=none (message not signed) header.i=none;
spf=Pass smtp.mailfrom=alice@sender.example;
spf=None smtp.helo=postmaster@mta1.sender.example;
dmarc=pass (p=quarantine dis=none) d=sender.example
```

From this header we can see that:

- The message was evaluated by `mta2.receiver.example`.
- The message did not pass DKIM as it was not signed.
- The message passed SPF using the domain found in the `smtp.mailfrom` or `From` header, `sender.example`.
- No SPF record was found for the domain used in the SMTP HELO message, `mta1.sender.example`.
- DMARC was evaluated using the policy of the domain (`d`) `sender.example`. Since the message passed SPF and the domains used for SPF and DMARC are aligned, the message passed DMARC and the

disposition or DMARC policy applied (`dis`) was `none`. We can also see that the domain has an enforcement policy (`p`) of `quarantine`.

Consider a more complex example incorporating ARC:

```
Authentication-Results:
mta3.receiver.example;
dkim=pass header.i=@sender.example header.s=selector5
header.b=B4f6tR4R;
arc=pass (i=1 spf=pass spfdomain=sender.example dkim=pass
dkdomain=sender.example dmarc=pass fromdomain=sender.example);
spf=pass (receiver.example: domain of bob@sender.example designates
192.0.2.1 as permitted sender) smtp.mailfrom=bob@sender.example;
dmarc=pass (p=NONE sp=NONE dis=NONE) header.from=sender.example
```

From this header we can see that:

- The message was evaluated by `mta3.receiver.example`.
- The message passed DKIM using the public key associated with the domain (`header.i`) `sender.example` and the selector (`header.s`) `selector5`. This key is published in a DNS TXT record located at `selector5._domainkey.sender.example`. The `header.b` field refers to the hash value found in the `DKIM-Signature` header.
- The message passed ARC relying on instance number 1 (`i=1`) of the results in the ARC chain. The remaining content summarizes the information found in the corresponding `ARC-Authentication-Results` header.
- The message passed SPF using the domain found in the `smtp.mailfrom` or `From` header, `sender.example`. In the "reason" section contained in parentheses, we can see that `receiver.example` found that IP address `192.0.2.1` was authorized by `sender.example`.
- DMARC was evaluated using the policy of the `header.from` domain, `sender.example`. Since the message passed at least one of SPF or DKIM, and the domains used were aligned, the message passed DMARC and the disposition (`dis`) was `NONE`. We can also see that the domain has an enforcement policy (`p`) of `NONE` and a subdomain policy (`sp`) of `NONE`.

C.1.2 DKIM-Signature

The `DKIM-Signature` header is a cryptographic signature that allows a receiving system to verify that the body of a message and certain specified headers have not been modified in transit. A receiving system can validate a message's authenticity using the message's contents and the specified public key. Table 5 describes the tags that are used as part of this header.

Table 5: DKIM-Signature Tags

Tag	Description

a	The cryptographic algorithm used to generate the signature.
b	The Base64-encoded hash of the headers listed in the h tag.
bh	The Base64-encoded hash of the message body.
c	Optional tag to indicate which level of modifications are to be tolerated, such as adjusting whitespace or line wrapping. Contains two separate values for the headers and message body in the form "[headers]/[body]", and valid values for each entry are "simple" and "relaxed", e.g. "relaxed/relaxed".
d	The domain where the public key has been published, used with the selector (s) to determine a DNS location of the form selector._domainkey.domain.example.
h	A list of headers that have been used to compute the header hash value, b.
i	Optional tag to associate a message with a specific identity (i.e. address or mailbox). The domain used must match the domain from the d tag.
s	The selector where the public key has been published, used with the domain (d) to determine a DNS location of the form selector._domainkey.domain.example.
t	Optional timestamp, in epoch UTC format (seconds since 1 January 1970).
v	Version of the specification, always "v=1".
x	Optional expiry timestamp, in epoch UTC format (seconds since 1 January 1970). Must be greater than the value of t.

C.1.3 Received-SPF

The `Received-SPF` header records SPF authentication results and may include additional information not found in the SPF section(s) of the `Authentication-Results` header. Table 6 describes the possible results of SPF validation, and Table 7 describes the fields that may be included in this header. This header may also include a "reason" section enclosed in parentheses with additional information on how the result was arrived at.

Table 6: Received-SPF Results

Result	Description
`fail`	The IP address was found to be unauthorized, usually by an SPF record terminated with "`-all`".
`neutral`	The IP address was not authorized, but the SPF record terminated with "`?all`". This result is typically handled equivalent to `pass`.
`none`	No SPF record was found for the `From` domain, nor for the SMTP HELO domain if it was also queried, or the `Return-Path` domain if such a header was present.
`pass`	The IP address was found to be authorized.
`permerror`	The SPF record could not be properly interpreted, like due to a syntax error or because the limit of 10 DNS lookups was exceeded.
`softfail`	The IP address was not authorized, but the SPF record terminated in "`~all`". This result is typically handled equivalent to `pass`.
`temperror`	An SPF record could not be retrieved, likely due to a DNS error.

Table 7: Received-SPF Fields

Field	Description
`client-ip`	The IP address of the sending SMTP client.
`envelope-from`	The address used in the `From` header.
`helo`	The host name given in the SMTP HELO command.
`identity`	The identity that was used in the validation, typically either "`mailfrom`" or "`helo`".
`mechanism`	The portion of the SPF record that matched, e.g. "`ip4:192.0.2.1`"
`problem`	If an error was returned, details about the error.
`receiver`	The hostname of the SPF verifier.

C.2 DMARC Aggregate Reports

DMARC aggregate reports provide system owners with information on messages that have been sent with one of their domains in the `From` field. These reports are typically sent by receiving systems once every 24 hours, but

some receivers may provide them more frequently. Reports are delivered via email to the address specified in a DMARC record's `rua` tag. The report is included as an attachment, compressed using either ZIP or GZIP.

A DMARC aggregate report is encoded using XML to facilitate automated processing and structured as specified in RFC 7489 [3]. The sections of the report, along with example excerpts, are described below.

C.2.1 Report Metadata

The `report_metadata` section contains information related to the overall report, including:

- The name (`org_name`) and an email address (`email`) for the originating organization;
- Extra contact information (`extra_contact_info`), such as a website address;
- A report identifier (`report_id`), which should be unique for a given reporting organization; and
- A `date_range` section with the beginning and ending timestamps of the period covered by the report, in Unix epoch seconds format.

An example `report_metadata` section is included below.

```
<report_metadata>
  <org_name>receiver.example</org_name>
  <email>noreply-dmarc@receiver.example</email>
  <extra_contact_info>https://receiver.example/dmarc</extra_contact_info>
  <report_id>75896041237538053212</report_id>
  <date_range>
    <begin>1602633600</begin>
    <end>1602719999</end>
  </date_range>
</report_metadata>
```

C.2.2 Policy Published

The `policy_published` section details the DMARC policy that the receiving system retrieved from DNS and used in its evaluation. This section includes the following information:

- `domain` is the domain portion of policy's DNS location, e.g. `domain.example` for the record found at `_dmarc.domain.example`.
- The remaining fields reflect either the explicit or default values of the corresponding tags from the DMARC policy record, including:
  - `adkim` – DKIM alignment, default `r` for relaxed.
  - `aspf` – SPF alignment, default `r` for relaxed.
  - `p` – Domain policy, must be explicitly `none`, `quarantine`, or `reject`.
  - `sp` – Subdomain policy, can optionally be either `none`, `quarantine`, or `reject`. Defaults to value of `p`.
  - `pct` – Percentage, default `100`.

An example `policy_published` section is included below.

```
<policy_published>
  <domain>owner.example</domain>
  <adkim>r</adkim>
  <aspf>r</aspf>
  <p>none</p>
  <sp>none</sp>
  <pct>100</pct>
</policy_published>
```

C.2.3 Records

An aggregate report will contain one or more `record` sections that detail the authentication and delivery results for a group of similar messages.

Each `record` contains one or more `row` sections that include the following:

- `source_ip`: The IP address of the server from which the messages were received. This is also the IP address that the reporter used for SPF authentication.
- `count`: The number of messages that were received with similar characteristics and processed similarly.
- `policy_evaluated`: Includes the authentication results for SPF, DKIM, and DMARC in several sections:
  - `disposition`: The DMARC policy applied by the receiver, either `none`, `quarantine`, or `reject`.
  - `dkim`: The result of DKIM authentication including domain alignment, either `pass` or `fail`.
  - `spf`: The result of SPF authentication including domain alignment, either `pass` or `fail`.
  - `reason`: An optional section with `type` and `comment` fields. This section is typically used if the receiving system took an action other than that specified in the domain's DMARC policy, such as delivering a message that failed DMARC because it passed ARC validation.

The `identifiers` section typically includes a `header_from` field, which is the domain found in the `From` header. This is the domain that must align with the domains used for SPF and DKIM. This section may also include an `envelope_from` field, which includes the domain used in the SMTP HELO message. If the `envelope_from` field is not included, this information is likely available in the `spf` / `domain` field of the `auth_results` section.

The `auth_results` section contains two subsections detailing the authentication results for DKIM and SPF. Note that messages that have been forwarded can include more than one `auth_results` section, which can be useful in tracing a message's path.

The `dkim` subsection includes the following fields:

- `domain`: The domain specified by the `d` (domain) tag of the `DKIM-Signature` header.
- `result`: The result of validating a message's DKIM signature, typically either `pass` or `fail`.
- `selector`: The selector specified by the `s` (selector) tag of the `DKIM-Signature` header.

Note that if a message was not signed with DKIM, the `result` and `selector` fields may be empty, omitted, or populated with "`none`".

The `spf` subsection includes the following fields:

- `domain`: The domain used in SPF authentication.
- `result`: The result of SPF authentication, most commonly either `pass`, `fail`, or `none` (see Table 6 for the full set of possible results).
- `scope`: Optionally included, this field specifies the source of the domain used for SPF authentication, either `helo` for the SMTP HELO message or `mfrom` for the `From` header. Assumed to be `mfrom` if not specified.

In the example record below, the message:

- Passed SPF for the SMTP HELO domain but failed alignment because this domain did not match the `header_from` domain;
- Passed DKIM including alignment; and
- Passed DMARC and was delivered (`disposition` of `none`).

```
<record>
  <row>
    <source_ip>192.0.2.1/source_ip>
    <count>1</count>
    <policy_evaluated>
      <disposition>none</disposition>
      <dkim>pass</dkim>
      <spf>fail</spf>
    </policy_evaluated>
  </row>
  <identifiers>
    <header_from>owner.example</header_from>
  </identifiers>
  <auth_results>
    <dkim>
      <domain>owner.example</domain>
      <result>pass</result>
      <selector>2021-02-01</selector>
    </dkim>
    <spf>
      <domain>third-party.example</domain>
      <result>pass</result>
    </spf>
  </auth_results>
</record>
```

In the example below, three messages:

- Passed SPF and DKIM for the domain `forwarder.example`;

- Failed DMARC because the domains used for both SPF and DKIM did not align with the `header_from` domain, `owner.example`; and
- Delivered (ultimately) by the receiver (`disposition` of `none`) because they passed ARC validation (noted in the `reason`/`comment` field as `arc=pass`).
  - This means that the receiving system relied on the ARC signature applied by `forwarder.example` attesting that the messages had passed SPF and DKIM before being forwarded.

```
<record>
  <row>
    <source_ip>198.51.100.1</source_ip>
    <count>3</count>
    <policy_evaluated>
      <disposition>none</disposition>
      <dkim>fail</dkim>
      <spf>fail</spf>
      <reason>
        <type>local_policy</type>
        <comment>arc=pass</comment>
      </reason>
    </policy_evaluated>
  </row>
  <identifiers>
    <header_from>owner.example</header_from>
  </identifiers>
  <auth_results>
    <dkim>
      <domain>forwarder.example</domain>
      <result>pass</result>
      <selector>forward-dkim</selector>
    </dkim>
    <spf>
      <domain>forwarder.example</domain>
      <result>pass</result>
    </spf>
  </auth_results>
</record>
```

C.3 Interpreting DMARC Results

Manually analyzing even a moderate number of aggregate reports is generally not feasible. There are many open source, free, and commercial tools that can assist you by automatically processing these reports, extracting the relevant information, and presenting summarized data for review and analysis.

C.3.1 SPF Results

SPF results can be diagnosed by examining the contents of the `auth_results` / `spf` / `result` and `policy_evaluated` / `spf` fields. The possible combinations of results and their meanings are summarized in Table 8.

Table 8: Meaning of SPF Results

Auth Results	Policy Evaluated	Meaning
Other than `pass`	`fail`	SPF missing / error / failure
`pass`	`fail`	SPF misaligned
`pass`	`pass`	SPF aligned

For SPF errors or failures, examine the `auth_results` / `spf` / `domain` field to determine the domain used. Ensure that an SPF record for this domain exists, it is properly formatted, and it includes the correct set of IP addresses.

For SPF misalignment, examine the `auth_results` / `spf` / `domain` field and compare it to the `header_from` field. If the SPF domain is a subdomain of the `header_from` domain, the misalignment may be caused by a DMARC policy requiring strict alignment for SPF (`aspf=s`). This can be corrected by one of the following actions:

- Creating a DMARC record specific to the subdomain;
- Changing the domain use; or
- Changing the alignment policy of the organizational domain to relaxed.

If it is not a subdomain, the misalignment may be caused by a third-party sender. In some cases, third-party senders support the use of a custom `Return-Path` address to attend to this issue.

C.3.2 DKIM Results

DKIM results can be diagnosed by examining the contents of the `auth_results` / `dkim` / `result` and `policy_evaluated` / `dkim` fields. The possible combinations of results and their meanings are summarized in Table 9.

Table 9: Meaning of DKIM Results

Auth Results	Policy Evaluated	Meaning
Other than `pass`	`fail`	DKIM missing / error / failure
`pass`	`fail`	DKIM misaligned
`pass`	`pass`	DKIM aligned

If DKIM is missing, ensure that it has been deployed correctly for the sending server.

For DKIM errors or failures, examine the `auth_results` / `dkim` / `domain` and `selector` fields to determine the DKIM DNS record used. Ensure that this DKIM record exists, it is properly formatted, and the servers that should use it are correctly configured.

For DKIM misalignment, examine the `auth_results` / `dkim` / `domain` field and compare it to the `header_from` field. If the DKIM domain is a subdomain of the `header_from` domain, the misalignment may be caused by a DMARC policy requiring strict alignment for DKIM (`adkim=s`). This can be corrected by:

- Creating a DMARC record specific to the subdomain;
- Changing the domain used; or
- Changing the alignment policy of the organizational domain to relaxed.

If it is not a subdomain, the misalignment may be caused by a third-party sender. In some cases, this can be corrected by creating a CNAME record that points to a DKIM record maintained by the third party.

C.3.3 DMARC Failures

If both SPF and DKIM fail, either outright or because of domain misalignment, a message fails DMARC and the specified enforcement policy is applied. This is indicated by a value of `fail` in both the `policy_evaluated` / `spf` and `policy_evaluated` / `dkim` fields. You should monitor your DMARC report data for unusual spikes in the number of DMARC failures, which can indicate a problem with your domain.

DMARC failures are typically caused by the following:

- **Incomplete deployment / misconfiguration**: A sending server's IP address has not been included in the SPF record for the domain, DKIM has not been deployed, and/or a policy of strict domain alignment has not been met.
- **Forwarding**: Messages that are forwarded by an intermediate server (e.g. a mailing list) often fail DMARC when subsequently received. Because SPF is more susceptible to this problem than DKIM, it can be partially mitigated by implementing DKIM. If it is supported, having the forwarder set a custom `Return-`

`Path` header or replace the `From` header can also address SPF issues. ARC also provides a mechanism for forwarded messages to be trusted but is not widely deployed.

- **Third parties**: Third-party senders that have not been correctly configured to support SPF and DKIM are indistinguishable from unauthorized senders who are sending spoofed messages. DMARC reports can help to identify third-party services that are being used within your organization, allowing for those that have been approved to be properly configured. Third-party senders that cannot fully support SPF (including a custom `Return-Path` header), DKIM, and DMARC should be avoided where possible.
- **Spoofing**: DMARC failures that cannot be attributed to misconfigurations, forwarding, or third-party senders are likely attributable to spoofing.

C.3.4 Spoofing Campaign Characteristics

The spoofing activity that we have observed targeting domains belonging to the Government of Canada has exhibited the following characteristics:

- **Commodity spoofing**: We have observed a continuous low level of spoofing affecting seemingly any domain with a web or DNS presence. We do not assess this activity to be targeted, and it should be expected for any domain.
- **Short duration**: Targeted campaigns have tended to be confined to a time span of a few days, with the message volume typically resembling a curve peaking in the middle.
- **Dispersed infrastructure**: Targeted campaigns are typically spread across many servers from a variety of hosting providers, networks, and geographic areas.
- **Related identifiers**: In some cases, activity can be clustered into campaigns based on the use of common or related identifiers, such as the domains used in the servers' SMTP HELO messages or the `From` header.