Overview – dmarc.org

Archived: 2026-04-05 16:13:07 UTC

Background

Email authentication technologies SPF and DKIM were developed over a decade ago in order to provide greater assurance on the identity of the sender of a message. Adoption of these technologies has steadily increased but the problem of fraudulent and deceptive emails has not abated. It would seem that if senders used these technologies, then email receivers would easily be able to differentiate the fraudulent messages from the ones that properly authenticated to the domain. Unfortunately, it has not worked out that way for a number of reasons.

- Many senders have a complex email environment with many systems sending email, often including 3rd party service providers. Ensuring that every message can be authenticated using SPF or DKIM is a complex task, particularly given that these environments are in a perpetual state of flux.
- If a domain owner sends a mix of messages, some of which can be authenticated and others that can’t, then email receivers are forced to discern between the legitimate messages that don’t authenticate and the fraudulent messages that also don’t authenticate. By nature, spam algorithms are error prone and need to constantly evolve to respond to the changing tactics of spammers. The result is that some fraudulent messages will inevitably make their way to the end user’s inbox.
- Senders get very poor feedback on their mail authentication deployments. Unless messages bounce back to the sender, there is no way to determine how many legitimate messages are being sent that can’t be authenticated or even the scope of the fraudulent emails that are spoofing the sender’s domain. This makes troubleshooting mail authentication issues very hard, particularly in complex mail environments.
- Even if a sender has buttoned down their mail authentication infrastructure and all of their legitimate messages can be authenticated, email receivers are wary to reject unauthenticated messages because they cannot be sure that there is not some stream of legitimate messages that are going unsigned.

The only way these problems can be addressed is when senders and receivers share information with each other. Receivers supply senders with information about their mail authentication infrastructure while senders tell receivers what to do when a message is received that does not authenticate.

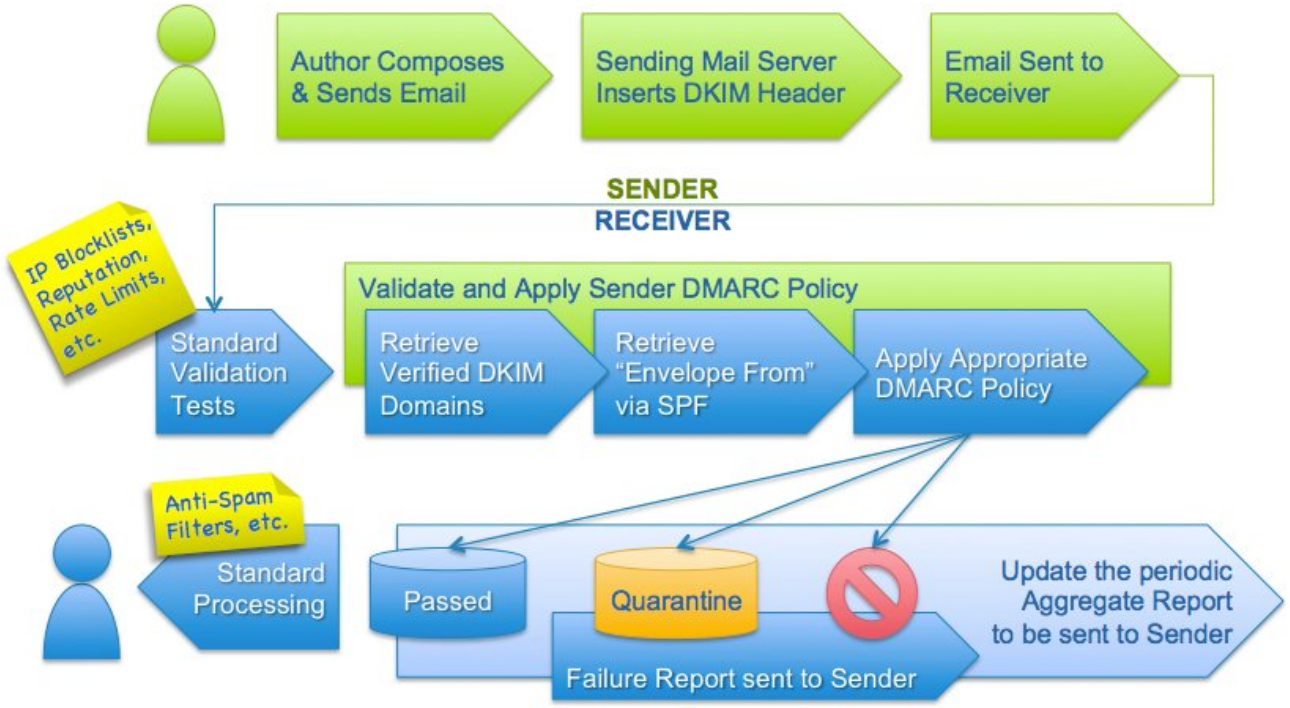
In 2007, PayPal pioneered this approach and worked out a system with Yahoo! Mail and later Gmail to collaborate in this fashion. The results were extremely effective, leading to a significant decrease in suspected fraudulent email purported to be from PayPal being accepted by these receivers.

The goal of DMARC is to build on this system of senders and receivers collaborating to improve mail authentication practices of senders and enable receivers to reject unauthenticated messages.

DMARC and the Email Authentication Process

DMARC is designed to fit into an organization’s existing inbound email authentication process. The way it works is to help email receivers determine if the purported message “aligns” with what the receiver knows about the

sender. If not, DMARC includes guidance on how to handle the “non-aligned” messages. For example, assuming that a receiver deploys SPF and DKIM, plus its own spam filters, the flow may look something like this:



In the above example, testing for alignment according to DMARC is applied at the same point where ADSP would be applied in the flow. All other tests remain unaffected.

At a high level, DMARC is designed to satisfy the following requirements:

- Minimize false positives.
- Provide robust authentication reporting.
- Assert sender policy at receivers.
- Reduce successful phishing delivery.
- Work at Internet scale.
- Minimize complexity.

It is important to note that DMARC builds upon both the DomainKeys Identified Mail (DKIM) and Sender Policy Framework (SPF) specifications that are currently being developed within the IETF. DMARC is designed to replace ADSP by adding support for:

- wildcarding or subdomain policies,
- non-existent subdomains,
- slow rollout (e.g. percent experiments)
- SPF
- quarantining mail

Anatomy of a DMARC resource record in the DNS

DMARC policies are published in the DNS as text (TXT) resource records (RR) and announce what an email receiver should do with non-aligned mail it receives.

Consider an example DMARC TXT RR for the domain “sender.dmarcdomain.com” that reads:

```
"v=DMARC1;p=reject;pct=100;rua=mailto:postmaster@dmarcdomain.com"
```

In this example, the sender requests that the receiver outright reject all non-aligned messages and send a report, in a specified aggregate format, about the rejections to a specified address. If the sender was testing its configuration, it could replace “reject” with “quarantine” which would tell the receiver they shouldn’t necessarily reject the message, but consider quarantining it.

DMARC records follow the extensible “tag-value” syntax for DNS-based key records defined in DKIM. The following chart illustrates some of the available tags:

Tag Name	Purpose	Sample
v	Protocol version	v=DMARC1
pct	Percentage of messages subjected to filtering	pct=20
ruf	Reporting URI for forensic reports	ruf=mailto:authfail@example.com
rua	Reporting URI of aggregate reports	rua=mailto:aggrep@example.com
p	Policy for organizational domain	p=quarantine
sp	Policy for subdomains of the OD	sp=reject
adkim	Alignment mode for DKIM	adkim=s
aspf	Alignment mode for SPF	aspf=r

NOTE: *The examples in this chart are illustrative only and should not be relied upon in lieu of the specification. Please refer to the specification page for the most up-to-date and accurate version.*

How Senders Deploy DMARC in 5-Easy Steps

DMARC has been designed based on real-world experience by some of the world’s largest email senders and receivers deploying SPF and DKIM. The specification takes into account the fact that it is nearly impossible for an organization to flip a switch to production. There are a number of built-in methods for “throttling” the DMARC processing so that all parties can ease into full deployment over time.

1. Deploy DKIM & SPF. You have to cover the basics, first.
2. Ensure that your mailers are correctly aligning the appropriate identifiers.
3. Publish a DMARC record with the “none” flag set for the policies, which requests data reports.
4. Analyze the data and modify your mail streams as appropriate.
5. Modify your DMARC policy flags from “none” to “quarantine” to “reject” as you gain experience.

Source: https://dmarc.org/overview