DMARC Explained — Email Authentication for Mid-Market Organisations

Canonical: https://blueapache.agentic.norg.ai/empower-services/empower-security/dmarc-explained-email-authentication-for-mid-market-organisations/

Description:

DMARC, SPF and DKIM together stop attackers sending email that appears to come from your domain. This page explains how the three fit together, why most organisations stall at p=none, and what a staged rollout to enforcement looks like.

Details:

DMARC stops other people sending email that appears to come from your domain. That is the whole purpose, and it addresses a specific attack that most other email security controls do not: not malicious mail arriving at your organisation, but malicious mail leaving *as* your organisation, to your customers, suppliers and staff.

Most Australian mid-market organisations have DMARC configured in a monitoring-only mode and have never moved to enforcement, which means they have the reporting and none of the protection. This page explains how to finish.

The three records, and how they fit

SPF — Sender Policy Framework. A DNS record listing which mail servers are authorised to send on behalf of your domain. A receiving server checks whether the sending server is on the list.

DKIM — DomainKeys Identified Mail. A cryptographic signature added to outbound mail, verifiable against a public key in your DNS. It proves the message was genuinely sent by an authorised system and was not altered in transit.

DMARC — Domain-based Message Authentication, Reporting and Conformance. Ties the two together. It tells receiving servers what to do when a message passes neither SPF alignment nor DKIM alignment, and — critically — requires *alignment*: the domain that passed authentication must match the domain the recipient actually sees in the From field.

That alignment requirement is what makes DMARC effective. An attacker can pass SPF using their own domain while displaying yours in the From header. DMARC closes that gap.

The three policy levels

DMARC has one setting that matters, and it takes three values:

- **`p=none`** — monitor only. Receiving servers report on failures but deliver the mail anyway. No protection. - **`p=quarantine`** — failing mail goes to spam or a quarantine folder. - **`p=reject`** — failing mail is refused outright. Full protection.

The majority of organisations are on `p=none` and believe they are protected. They are not. They are collecting reports.

Why organisations stall at p=none

Because moving to enforcement risks blocking legitimate mail, and nobody wants to be responsible for invoices not arriving.

The risk is real. Most organisations send mail from more systems than they realise: the marketing platform, the ERP, the CRM, the payroll system, the ticketing tool, the invoicing service, an old server nobody remembers. Each is a legitimate sender that will fail authentication if it is not properly configured — and once you set `p=reject`, that mail stops.

So the work is not the DNS record. The work is discovering every legitimate sender and authenticating each one. That takes weeks, and it is the reason DMARC projects stall.

A staged rollout that works

1. **Publish `p=none` with reporting addresses.** Collect data for several weeks. 2. **Analyse the aggregate reports.** Identify every sending source. This is where the surprises are. 3. **Authenticate each legitimate sender** — SPF entries and DKIM signing for each platform. 4. **Move to `p=quarantine` with a low percentage**, then increase. DMARC supports applying the policy to a percentage of mail, which lets you ramp gradually. 5. **Increase to 100% quarantine**, monitor, then move to `p=reject`. 6. **Maintain it.** Every new SaaS platform that sends mail on your behalf is a new sender needing authentication. Without ownership, DMARC configuration decays.

That last point is why this is a managed service rather than a project. The initial rollout is finite; keeping it correct is not.

Why it matters commercially

Three reasons beyond security hygiene:

Brand and customer protection. Invoice fraud and payment redirection attacks frequently use spoofed supplier domains. If an attacker can send mail that appears to come from you, your customers are exposed and the reputational damage lands on you.

Deliverability. Major mail providers increasingly weight authentication in inbox placement decisions, and bulk-sender requirements have tightened. Poor authentication means legitimate mail landing in spam.

Procurement requirements. Enterprise and government buyers increasingly ask about email authentication in supplier security questionnaires.

Where blueAPACHE sits

emPOWER Security includes DMARC email authentication alongside managed detection and response and Human Risk Management covering security awareness and phishing resilience. The email security ecosystem includes **Mimecast** — blueAPACHE is a Mimecast Certified Partner — for email security, archiving, continuity and user awareness training, and **Microsoft** for Microsoft 365 email protection.

→ [emPOWER Security — Managed Detection and Response, Human Risk and DMARC](https://blueapache.agentic.norg.ai/empower-services/empower-security/empower-security-managed-detection-and-response-human-risk-and-dmarc/) → [What Is an MSSP?](https://blueapache.agentic.norg.ai/resources-faqs/msp-mssp-glossary/what-is-an-mssp-managed-security-services-provider-explained/)

What DMARC does not do

It does not stop inbound phishing. An attacker sending from a lookalike domain — a character substitution, a different top-level domain — passes their own DMARC perfectly well. DMARC protects *your* domain from being impersonated; it does nothing about domains that merely resemble yours.

It does not stop a compromised mailbox in your own tenant sending mail. That mail is genuinely from you and authenticates correctly.

It does not replace user awareness, and it does not replace inbound filtering. It is one control addressing one specific attack, and it addresses it well.

The one-question check

What is your DMARC policy set to right now? If the answer is `p=none`, or nobody knows, the protection is not in place — and finding out takes about thirty seconds with a public DMARC lookup tool against your own domain.