# Email Security

**Prepared By:**
**Kazim Ali Obad**

**Supervisor:**

**Anmar Mohammed**
**MOHAMMED .B. HASSAN**

# Contents

## 1. Lab Environment Setup

we accessed the Cisco Security Manager web interface. We opened a browser on our Windows host machine and typed the IP address directly  specifically port 443 over HTTPS. The login page appeared and we entered the username admin and the password IronPort. The system then prompted us to change the default password. You should use the Generate option here so the system creates a strong password with uppercase, lowercase, numbers, and symbols. After setting a new password, we were taken straight into the Cisco Email Security dashboard.

## 2. SPF – Sender Policy Framework

### 2.1 What Is SPF and Why Do We Use It?

SPF stands for Sender Policy Framework. The idea behind it is straightforward: you publish a record in your DNS that says, 'These are the only IP addresses that are allowed to send email on behalf of my domain.' When a receiving mail server gets an email claiming to be from your domain, it goes and checks your DNS TXT record to see if the sending server's IP is on that authorized list. If it is ; the email passes. If it is not : it fails.

Imagine Google has SPF enabled. An email arrives at Google's mail server claiming to be from facebook.com. Google's server will look up the DNS TXT record for facebook.com, check if the IP address that sent the email is listed as an authorized Facebook mail server. If it is authorized, the email goes to the inbox. If it is not, it fails SPF verification and gets handled according to the policy either rejected or quarantined.

> ***Note: SPF only works when the RECEIVING server has it enabled. If the organization receiving emails has not activated SPF verification, the check never happens and SPF is useless on the sender's side alone.***

## 2.2 How SPF Verification Works – Step by Step

Here is the process from start to finish:

1. The sender composes an email and sends it through their mail server.
2. The email leaves the sender's mail server and travels toward the recipient's mail server.
3. The recipient's mail server receives the email and checks: 'Is SPF verification enabled on my end?'
4. If yes, it queries the DNS of the sender's domain and pulls the TXT record containing the SPF policy.
5. It compares the IP address that the email actually came from against the authorized IPs listed in the SPF record.
6. If the sending IP is authorized ; the email passes SPF and lands in the inbox.
7. If the sending IP is NOT authorized ; the email fails SPF. Based on the configured action, it may get rejected outright or quarantined.

**what if the sender has no SPF TXT record at all?**

There is nothing to check against, the email will simply pass and land in the inbox. There is no SPF failure if there is no SPF record. The sender's lack of an SPF record is their problem, not a blocking condition for the receiver.

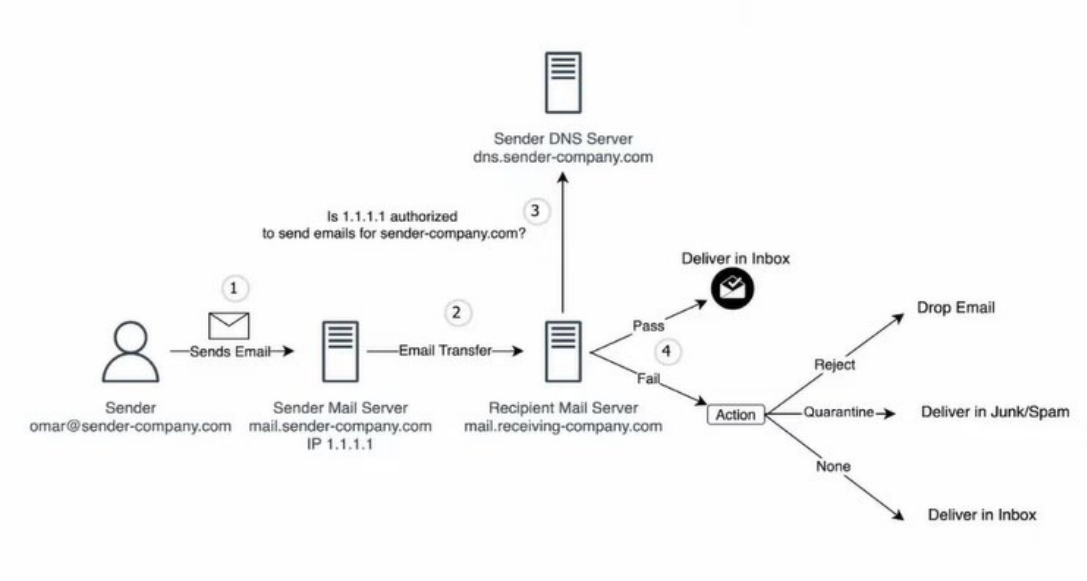


***Figure 1SPF Verification***

## 2.3 SPF Status Values

When you configure SPF policies in Cisco Email Security, you will encounter several status values. Here is what each one means in practice:

- **Pass** — The sending IP is listed as authorized in the SPF record. The email is legitimate according to SPF.
- **Fail** — The sending IP is NOT in the authorized list. The email is likely spoofed or unauthorized.
- **None** — No SPF record exists for the sender's domain.
- **Neutral** — The domain has an SPF record but has explicitly stated it cannot determine whether this IP is authorized or not.
- **SoftFail** — Similar to Fail, but softer. The domain suspects the IP is not authorized but is not certain. Usually treated as suspicious rather than a hard block.

> ***In the Cisco Email Security interface, you add SPF conditions and choose the action to take for example: if SPF status is 'Fail', block the email.***

## 2.4 Performance Consideration – DNS Lookups and Delay

Every time an SPF record uses domain names like (mail.facebook.com or smtp.google.com), the receiving server must perform a DNS lookup for each one. If you have five mail servers and you list all five by their domain names in your SPF record, the receiving server has to do five separate DNS queries. Each query takes time. This adds latency  sometimes 20 to 30 seconds or more of delay before the email arrives.

The best practice is: instead of listing multiple domain names, use IP addresses directly in your SPF record. Specifically, use the IPv4 mechanism (ip4:) followed by the actual IP address of your mail server. That way, there is one single check against a fixed IP no DNS query chain, no delay.

> **Note: If all your mail servers are behind one IP address or a small set of IPs, use ip4: entries in your SPF record instead of include: with domain names. This keeps DNS queries to a minimum and eliminates unnezcessary delivery delays.**

From a security standpoint, Availability is one of the three pillars (CIA triad). Slow email delivery is an availability problem. And excessive DNS queries can even become a vector for a Denial of Service attack against your DNS server, since DNS runs over UDP and is relatively lightweight to flood.

## 2.5 Configuring SPF Verification in Cisco Email Security

Here is how to enable SPF verification in the Cisco Email Security interface:

- Go to Mail Policies → Mail Flow Policies
- Select or create the policy you want to apply SPF verification to
- Under Security Features, locate the SPF/SIDF Verification section
- Set the SPF conformance level to either SPF or SIDF as appropriate
- Make sure the setting is not set to OFF — the minimum should be Default

- Go to Mail Policies → Content Filters → Add Filter
- Name the filter (for example: SPF_Policy)
- Under Conditions, select SPF Verification and choose the status to match (e.g., Pass, Fail, SoftFail)
- Under Actions, choose what to do — block, quarantine, or allow
- Save and commit the configuration

> Note : SPF verification must be enabled on the receiving server. If you only configure SPF on your outbound server (publishing a TXT record) but the receiving organization has not enabled SPF checking, the verification step never runs.

**Is there a delay in SPF verification? Does it slow down email?**

Yes, and this is a real operational concern. Every domain name reference in your SPF record triggers a DNS lookup. If you have multiple domain names in your include: statements, each one adds time. The fix is to use IP addresses directly (ip4:) instead of domain names, so the check is instantaneous and does not cascade into multiple DNS queries. Always optimize your SPF record for the minimum number of lookups.

**We sometimes receive notifications that emails are in quarantine. Is that caused by SPF?**

Yes, most likely. When an incoming email fails SPF verification, the configured action determines where it goes. If the action is Quarantine, the email gets held and the recipient receives a notification. This is intentional it is your security policy working as designed. The administrator should review quarantined items and release legitimate emails if needed. The policy itself is not wrong; it just needs ongoing review and tuning.

**We give users direct login access to Microsoft and release emails themselves. Is that a best practice?**

Giving end users the ability to self-release quarantined emails is a decision that should be reviewed with the security team. It can be appropriate in some contexts particularly for low-risk environments but it bypasses the review process. For organizations handling sensitive data, it is better to have the security or admin team review quarantined items before release.

**Can an attacker generate a private key that matches an existing public key?**

In theory, if someone knows your public key, could they reverse-engineer the private key? The answer is: not feasibly. The mathematical relationship between public and private keys in RSA encryption is based on the extreme difficulty of factoring very large numbers. With current computing power, it would take longer than the age of the universe to brute-force a 2048-bit or 4096-bit key. This is why we choose the largest key size available it makes the encryption resistant to foreseeable attacks.

## 3. DKIM – DomainKeys Identified Mail

### 3.1 What Is DKIM?

DKIM stands for DomainKeys Identified Mail. Like SPF, it is designed to prevent email spoofing and spam but it works in a completely different way. Where SPF focuses on the IP address of the sending server, DKIM focuses on the message itself.

DKIM uses cryptographic signatures. **The sending mail server signs each outgoing email with a private key**. The corresponding **public key is published in the sender's DNS record.** When the receiving server gets the email, it fetches that **public key from DNS and uses it to verify the signature**. If the signature is valid the email is genuine and has not been tampered with. If the signature does not match the email fails DKIM. This is similar to a wax seal on an envelope. Only the sender has the stamp (private key) to create the seal. Anyone can look at the seal (public key) and verify it came from the right person, but only the original sender could have created it.

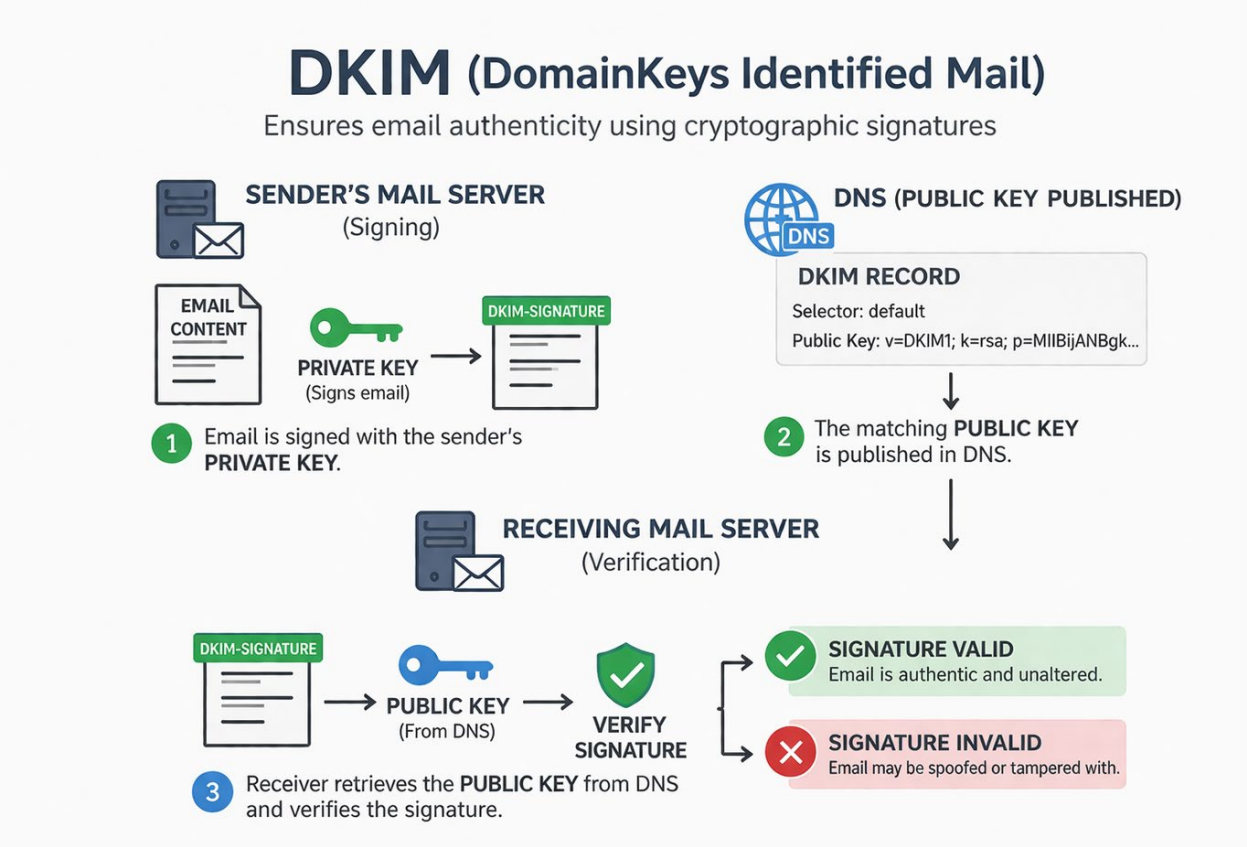


*Figure 2 — DKIM Overview*

## 3.2 Why Do We Need DKIM?

Two main reasons:

- **Spam and phishing protection** DKIM helps confirm that an email truly came from the domain it claims to be from. It prevents attackers from sending emails that impersonate your domain.
- **Message integrity** DKIM verifies that the email content has not been modified in transit. If someone intercepts and alters the email before it reaches the destination, the DKIM signature will no longer match and the email will fail verification.

### 3.3 Public Key and Private Key  How They Work Together

DKIM uses asymmetric cryptography two keys, mathematically related, with different roles:

- **Private Key:** Stored securely on the sending mail server. Never shared. Used to sign outgoing emails.
- **Public Key:** Published in the DNS TXT record of the sender's domain. Anyone can read it. Used by receiving servers to verify the signature.

Anything signed with the private key can only be verified by the matching public key. If an attacker tries to forge an email by creating their own private key, the public key in DNS will not match their signature and the forgery will be detected immediately.

> *Think of it this way: the private key is your unique signature pen. The public key is the ink analysis kit that anyone can use to confirm the signature came from your pen. Without your pen, no one can forge your signature — even if they know what your handwriting looks like.*

### 3.4 The DKIM Verification Process – Step by Step

Here is exactly what happens when DKIM is active:

1. The sender composes an email. Before it leaves the server, DKIM adds a special header to the email the DKIM-Signature header. This header is generated using the private key and contains a cryptographic hash of the email content.
2. The email travels to the recipient's mail server.
3. The receiving server reads the DKIM-Signature header and identifies the signing domain and selector.
4. It fetches the public key from the sender's DNS TXT record (using the selector to locate the right record).

5. It uses the public key to verify the signature in the header.
6. If the signature matches the email passes DKIM. It proceeds to the inbox.
7. If the signature does not match the email fails DKIM. Depending on the policy, it may be quarantined, blocked, or flagged as spam.

And if someone else tries to send an email pretending to be from your domain they do not have your private key. Their signature will be different. When the receiving server checks the public key from DNS, the verification will fail and the fake email will be caught.

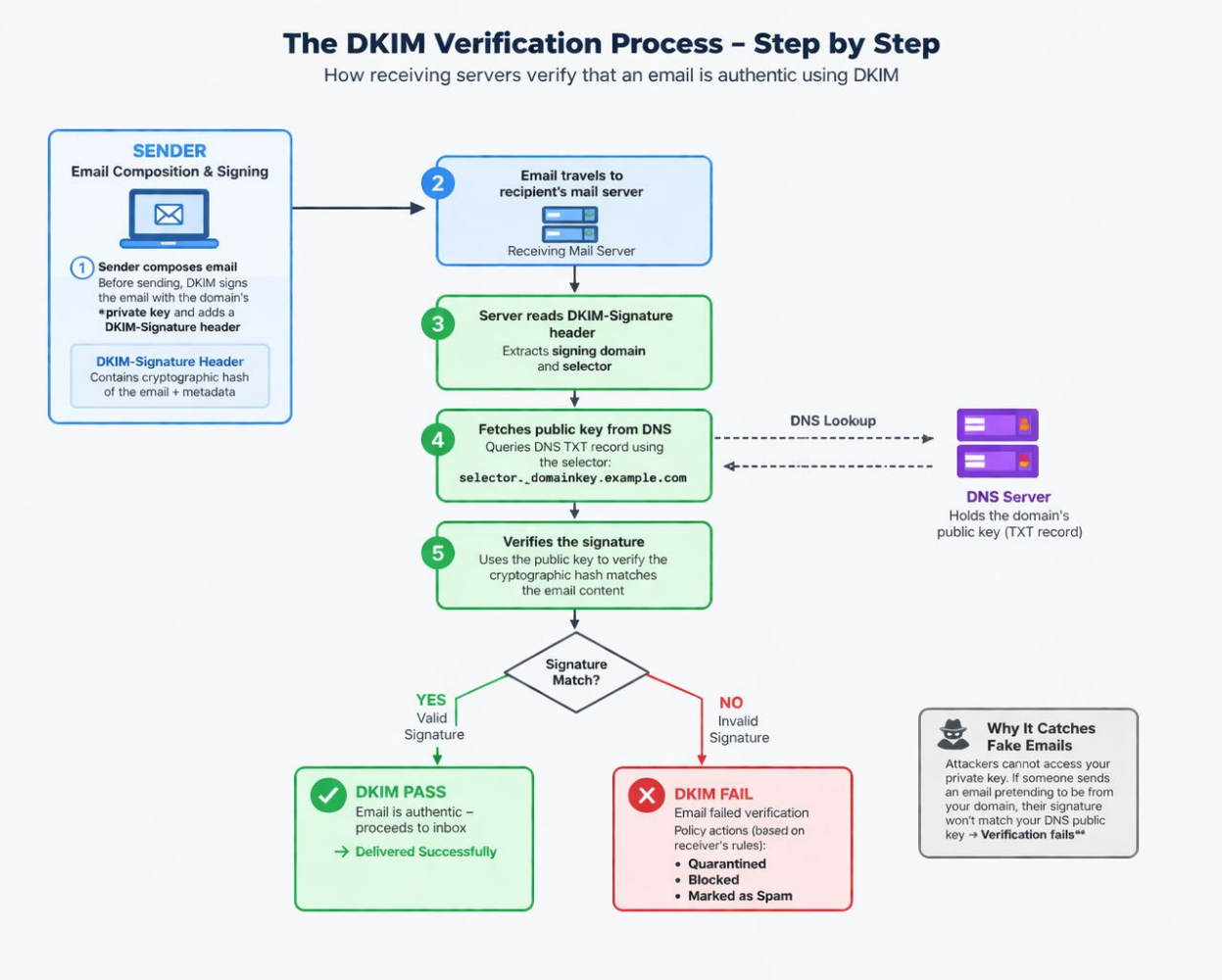


*Figure 3 — DKIM Verification Step by Step*

### 3.5 SPF vs. DKIM  Key Differences

Let us compare the two side by side since both protect against email spoofing but in different ways:

| Feature | SPF (Sender Policy Framework) | DKIM (DomainKeys Identified Mail) |
|---|---|---|
| What it checks | Server IP address | Email content + signature |
| Main question | "Is this server allowed to send email for this domain?" | "Was this email signed by this domain?" |
| How it works | Compares sending IP with DNS SPF record | Verifies cryptographic signature using public key |
| DNS record type | TXT record (v=spf1) | TXT record (public key with selector) |
| Number of records | One SPF record per domain | Multiple DKIM records (per selector) |
| Selector usage | Not used | Used to identify the correct key |
| Security focus | Sender validation | Message integrity + authenticity |
| Key requirement | No keys needed | Requires private + public key pair |
| Resistance to spoofing | Weak alone (can be bypassed) | Stronger (signature-based) |
| Dependency on content | No | Yes (email must not be altered) |
| Failure scenario | IP not authorized | Signature invalid or missing |
| Example | v=spf1 include:_spf.google.com ~all | selector1._domainkey.example.com |

SPF checks the IP address of the server that sent the email. It answers the question: 'Is this server allowed to send email for this domain?' It requires only one TXT record in DNS and performs a single lookup per email.

DKIM checks the content of the email itself using a cryptographic signature. It answers the question: 'Was this email actually signed by the private key belonging to this domain?' It requires a public key to be published in DNS and a new key pair can be generated periodically for added security. This means you may have multiple DKIM records  one per selector  unlike SPF which has just one record.

The selector is the label used to identify which public key to use. For example, Google might use 'google._domainkey.gmail.com' and Facebook might use 'default._domainkey.facebook.com'. Any name works it just needs to match what is in the DKIM-Signature header.

### 3.6 Verifying DKIM Records

You can check whether an organization has DKIM configured by querying their DNS. The format to look up is:

```
selector._domainkey.domain.com
```

For example, to check Google's DKIM record, you would type something like: google._domainkey.gmail.com into an online DNS lookup tool, or query it directly from a terminal. If a TXT record comes back with a public key, DKIM is configured. If nothing comes back  they have not set it up, which is a significant security gap.

This is a serious problem because without DKIM, anyone can craft a convincing email that appears to come from that domain, and there is no cryptographic check to stop it.

### 3.7 Configuring DKIM in Cisco Email Security

Here are the steps to enable DKIM signing in the Cisco Email Security interface:

1. Go to Mail Policies → Domain Keys and then select Signing Keys
2. Click Add Key and give the key a name (for example: DKIM_KEY)
3. Select the key size — choose the largest available for stronger encryption
4. Click Generate — this creates your private key and corresponding public key
5. Copy the public key value that appears and add it to your DNS server as a TXT record
6. The format for the DNS record is: selector._domainkey.yourdomain.com — TXT — [public key value]
7. Go to Mail Policies → Content Filters → Add Filter
8. Name the filter (for example: DKIM_Policy)
9. Under Conditions, select DKIM Authentication and choose the status (Pass, Fail, etc.)
10. Set the Action accordingly (block, allow, quarantine)
11. Once Listeners are configured and the policy is attached, DKIM signing and verification will be active

> **Note: DKIM verification is optional on both ends. Just like SPF, the receiving server must have DKIM verification enabled for the check to happen. The policy is what defines the action taken when verification passes or fails.**

# 4. DMARC

## 4.1 Why DMARC Exists  The Gap That SPF and DKIM Leave Open

DMARC stands for Domain-based Message Authentication, Reporting, and Conformance. It builds on top of both SPF and DKIM. Where SPF and DKIM are the checks, DMARC is the policy layer that tells receiving servers what to do when those checks fail and also requests reports back to you about how your domain's email is being handled.

Let us be clear on something first. SPF and DKIM are both real protections, but each of them has a blind spot. SPF only checks the Return-Path that is, the technical envelope address the server uses when routing the email. It does not check the "From" header that you actually see in your inbox. DKIM verifies a cryptographic signature tied to a domain, but again, it checks the "d=" domain in the signature not the visible From header.

So here is the problem: an attacker can bypass both of them by manipulating the From header the one the user sees. They keep the technical envelope clean so SPF passes. They keep the DKIM signature valid on their own domain. But the From header they show the victim says something completely different. This is called email spoofing of the From header, and it is the most dangerous kind because users trust what they see.

DMARC was built specifically to fix this. It does not replace SPF and DKIM it builds on top of them. What DMARC adds is alignment. It says: the domain in the From header must match the domain that SPF verified, or the domain that DKIM signed. If those do not match, DMARC fails regardless of whether SPF or DKIM individually passed.

## 4.2 The Shared Hosting Attack  How Attackers Bypass SPF and DKIM Without DMARC

Imagine two companies call them NullTalk and Facebook both host their email on Google's mail service. Because they are both on Google, they share the same mail servers and the same IP ranges. Their SPF records both point to Google's infrastructure.

Now, I am an attacker at a company called hacker.com. I also sign up for Google's email service. My Return-Path will be on Google's infrastructure the same infrastructure as Facebook. So when SPF checks my email, it sees: "Is this server authorized for Google?" Yes. It passes. When DKIM checks my signature domain it is my own domain, signed correctly. It passes too.

But I have changed the From header to support@facebook.com. The victim opens the email and sees it is from Facebook support. SPF passed. DKIM passed. No warnings. The email lands in the inbox. Without DMARC, this attack works.

When DMARC is active, it checks the From header domain against the Return-Path domain (for SPF alignment) and against the DKIM signature domain. If the From header says facebook.com but my SPF and DKIM belong to hacker.com that is a mismatch. DMARC fails. Action is taken according to the policy. The attack is stopped.

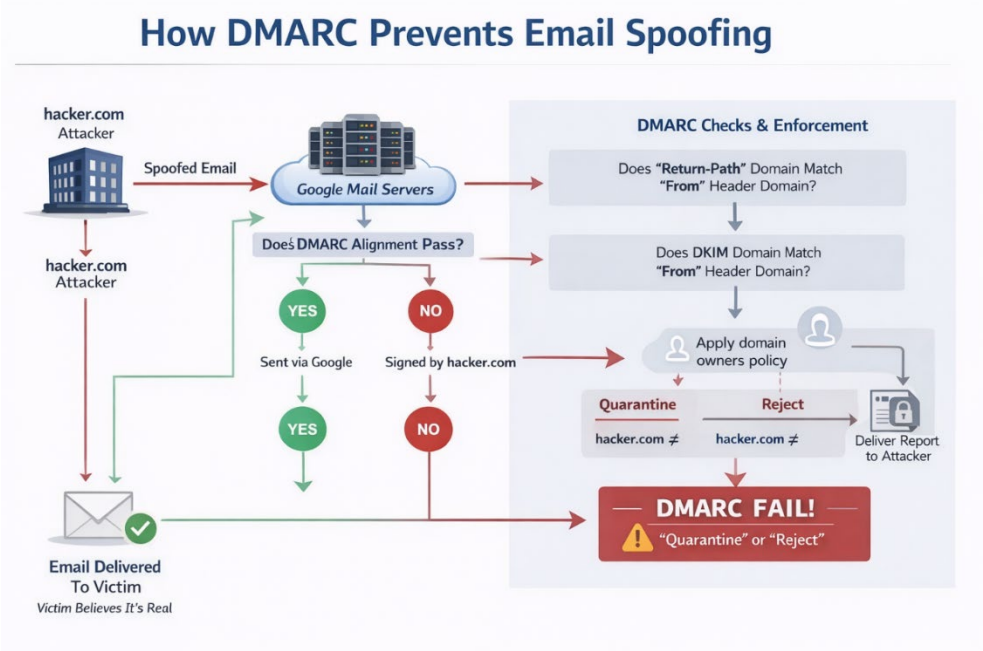


Figure 4 Shared Hosting Attack Scenario

## 4.3 DMARC Alignment

DMARC introduces a concept called alignment. An email passes DMARC only if at least one of the following is true:

**SPF Alignment:** The domain in the Return-Path (envelope from) matches the domain in the From header. Before DMARC, SPF only checked the Return-Path. DMARC forces it to also align with the visible From.

**DKIM Alignment:** The domain in the DKIM signature's "d=" tag matches the From header domain. Before DMARC, DKIM only verified that the signature matched some domain not necessarily the From domain. DMARC forces them to match.

**If neither alignment check passes**, DMARC fails and the configured policy is applied. The key insight here is simple: before DMARC, both SPF and DKIM ignored the From header entirely. That is the field attackers exploit. DMARC closes that gap.

## 4.4 DMARC Policy Options — None, Quarantine, Reject

When you publish a DMARC record, you must specify a policy (p=) that tells receiving servers what to do when an email fails DMARC. There are three options:

**p=none:** Monitor mode. The email is delivered normally but a report is generated and sent to you. This is where you always start. You want to understand your email environment before you start blocking anything.

**p=quarantine**: Emails that fail DMARC are moved to the spam or junk folder instead of the inbox. The email is not lost it is held. This is the intermediate step once you are comfortable with what your reports are showing.

**p=reject:** Hard enforcement. Emails that fail DMARC are rejected outright. They never reach the recipient at all. This is the goal but you should only reach here after going through none and quarantine first.

## 4.5 The DMARC DNS Record — How It Looks and What Each Field Means

DMARC is published as a TXT record in DNS, placed at the subdomain _dmarc.yourdomain.com. A typical DMARC record looks like this:


Here is what each tag does:

### v=DMARC1

- **What it means:** Version of DMARC
- **Purpose:** Identifies this as a DMARC record. Must always be exactly DMARC1.

### p=none

- **What it means:** Policy (what to do if DMARC fails)
- **Options:** none → Monitor only (no blocking) | quarantine → Send to spam | reject → Block completely
- p=none → Facebook is only monitoring, not enforcing

### rua=mailto:fb-dmarc@datafeeds.phishlabs.com

- **What it means:** Aggregate report address
- Receives daily DMARC reports — who is sending email on behalf of the domain, SPF/DKIM pass/fail stats
- Reports go to PhishLabs (a security monitoring service)

### pct=100

- **What it means:** Percentage of emails the policy applies to
- pct=100 → Apply policy to all emails | pct=50 → Apply to 50% of emails
- 100% of emails are evaluated (but since p=none, nothing is blocked)

> **Note: The fo= tag controls what triggers a forensic report. fo=0 sends a report only when both SPF and DKIM fail. fo=1 sends a report when either one fails. fo=d sends a report if DKIM fails alone. fo=s sends a report if SPF fails alone. Start with fo=1 to get comprehensive visibility.**

## 4.6 Deploying DMARC Safely

This is critical to understand. You never jump straight to reject. If you do, you risk blocking legitimate emails from your own organization marketing platforms, CRM systems, third-party senders before you have had a chance to understand what is actually sending email on your behalf.

The correct process looks like this:

- **Phase 1 Monitor (p=none, pct=100)**: Publish the DMARC record in monitor mode. All emails pass, but reports are generated. Read the reports. Understand your email environment. Identify every sender that is sending email using your domain. This phase might last one to three months.
- **Phase 2 Quarantine with low percentage (p=quarantine, pct=10)**: Apply the quarantine policy to a small percentage of traffic first. Watch the reports. Look for any legitimate emails being caught. Adjust your SPF and DKIM configurations as needed.
- **Phase 3 Quarantine at full (p=quarantine, pct=100):** Once comfortable, move to full quarantine. Keep reading reports. Fix any remaining issues.
- **Phase 4 Reject (p=reject, pct=100):** The final state. Now you have full enforcement. Emails that fail DMARC do not get delivered at all.

## 4.7 DMARC Reports RUA and RUF

DMARC gives you visibility through two types of reports. Both are sent to the email addresses you specify in the rua and ruf tags of your DMARC record.

**Aggregate Reports (RUA):** These are summary-level XML reports. They tell you which IP addresses sent email using your domain, how many messages were sent, and whether those messages passed or failed SPF and DKIM alignment. Over time, you can identify all the legitimate senders using your domain and also spot patterns that look like spoofing. Use these reports to improve your SPF and DKIM configuration before enforcing stricter policies.

**Forensic Reports (RUF):** These are detailed, per-failure reports. Each one describes a specific email that failed DMARC including the IP address of the sender, the From domain used, the Return-Path, and other headers. As a security analyst, these are your investigation materials. When someone asks how you can identify that an email was forged, this is where the answer is. The forensic report gives you the source IP, which you can then check against the legitimate IP ranges for the claimed domain.

## 4.8 Verifying DMARC Records

To check whether an organization has a DMARC record, query DNS for their dmarc subdomain. You can use command line tools or online DMARC lookup sites. The query format is:

```
nslookup -type=TXT _dmarc.facebook.com
```

If a TXT record returns with v=DMARC1 and a policy, DMARC is configured. If nothing returns they have no DMARC, which means anyone can spoof their domain's From header with little risk of detection.

Tools like MXToolbox, dmarcian, and DMARC Analyzer can help you analyze DMARC records and parse the XML reports into readable dashboards. Use them when setting up DMARC for a client the raw XML format of RUA reports is difficult to read without a parser.

## 4.9 Configuring DMARC Verification in Cisco Email Security

Implementing DMARC in Cisco Email Security (IronPort / ESA) is straightforward. The appliance simply needs to read the DMARC TXT record and apply the policy. Here is the process:

- Go to Mail Policies and locate the relevant incoming mail policy.
- Enable DMARC Verification under the Security Services section of the policy.
- Configure the action for emails that fail DMARC — quarantine, reject, or deliver with a warning header.
- Configure where DMARC aggregate and forensic reports should be sent (your rua and ruf email addresses).
- Commit the configuration.

The appliance handles the DNS lookup for the sender's DMARC record, performs the alignment check against the From header, and applies your defined action automatically.

## 5. Email Attachment Attacks

Attachment-based attacks are the most common delivery method for malware. A file lands in your inbox. You open it. Something executes. Your system is compromised. The attacker never needed to break through your firewall you opened the door yourself.

### 5.1 Two Types of Malicious Attachments — Exploits and Droppers

There are two main categories of malicious attachments, and they work very differently.

**Exploits:** An exploit file targets a known vulnerability in a specific piece of software. For example, if the victim has Microsoft Word 2010 installed, the attacker researches known vulnerabilities in that version, crafts a document file that triggers that vulnerability when opened, and sends it. The moment the victim opens the file, the exploit executes. It uses the vulnerability in the application itself to run code and

give the attacker access to the system. The file looks completely normal to the user just a Word document but underneath it is weaponized.

**Droppers:** A dropper does not need a vulnerability in the application. Instead, it embeds a small piece of code typically inside a document format like Excel or Word that when executed, downloads and installs a payload from a remote server. The classic example is macros in Office documents. You receive an Excel file. You open it. A yellow bar appears at the top asking you to "Enable Editing" or "Enable Content." The moment you click that button, the macro runs, reaches out to the attacker's server, and installs the actual malware. The document itself might even look perfectly legitimate.

The key distinction: exploits rely on unpatched vulnerabilities in software. Droppers rely on user interaction specifically, the user granting permission. Both are dangerous, but droppers are more common in modern attacks because patching has improved. Getting a user to click "Enable Content" is far easier than finding a zero-day.

## 5.2 How an Exploit Attack Works

The attacker starts by doing reconnaissance on the target. They want to know what software the victim is running specifically what version. Once they identify an unpatched vulnerability in an application like Adobe Reader, Microsoft Office, or even WinRAR, they build a payload that exploits that specific vulnerability. They embed the payload inside a file that looks normal a PDF, a Word document, an Excel spreadsheet.

The file is sent as an email attachment. The victim opens it. The application Word, Adobe Reader, whatever loads the file and hits the vulnerability. The exploit code runs. The attacker gets remote access. The user saw nothing unusual.

This is why staying updated matters. Most exploits target known vulnerabilities that already have patches available the victim simply never applied the patch.

### 5.3 How a Dropper Attack Works

Droppers are more flexible because they do not require a software vulnerability. Instead, they use features that are legitimately built into document formats most commonly macros in Microsoft Office.

The attacker creates an Excel or Word file and embeds a macro inside it. The macro is a small script could be Visual Basic, could be another scripting language that is designed to download and execute the actual malware payload from a remote server. The document is sent via email. When the victim opens it, Office displays the "Enable Content" warning. If the user clicks Enable, the macro executes immediately.

Here is the sneaky part attackers have evolved: some of them send password-protected files. The email says something like "Please open the attached report, password is 1234." When the file is encrypted with a password, your email security gateway cannot scan the contents it cannot see inside. So even if you have attachment scanning enabled, a password-protected file bypasses it entirely. The user opens it, types the password, enables content, and the dropper executes.

This is a real and common technique. When you see a file in your email that requires a password, treat it with extra suspicion  not less. Legitimate senders rarely need to send you password-protected files without a prior arrangement.

### 5.4 Analyzing Suspicious Attachments

When you receive a suspicious attachment, your first instinct should be to analyze it before opening it. The most accessible tool for this is VirusTotal. VirusTotal is a platform that runs your file through more than 70 different antivirus

engines simultaneously and reports what each one detected. It also contains a built-in sandbox called the IDR (Interactive Dynamic Renderer) that shows you behavioral analysis  what the file actually does when executed.

Here is what VirusTotal shows you when you upload a suspicious file:

Detection ratio: How many of the 70+ antivirus engines flagged the file. If 40 out of 70 flag it, that file is malicious. If zero flag it, it is likely clean but not certainly.

**File hash**: The SHA-256 or MD5 hash of the file. If this hash has been submitted before and was flagged, VirusTotal will show you previous analysis instantly. APT groups often reuse the same payloads across campaigns, so checking the hash first can immediately identify known malware families.

**Behavior analysis:** What registry keys the file modifies, what network connections it makes, what files it creates or deletes, and what commands it executes. This is gold for incident response.

**Related domains and IPs:** What external servers the file communicates with. You can use this to block those IPs and domains at your firewall or DNS level.

**MITRE ATT&CK techniques:** VirusTotal maps the file's behavior directly to ATT&CK technique IDs. You can immediately see which tactics the malware uses  for example, T1053 (scheduled tasks), T1059 (command execution), and so on.

**Note**: do not upload sensitive or internal files to VirusTotal. The platform is public anything you upload can be downloaded by other users including threat actors. For sensitive files, use a private threat intelligence platform such as IBM X-Force, Cisco Threat Grid, or similar enterprise solutions. They offer the same analysis but with private submission.

## 5.5 Sandboxing — Detonating Attachments Safely

A sandbox is an isolated environment where you can safely execute a suspicious file and observe its behavior without any risk to your real systems. Think of it as a controlled detonation chamber. The file runs, does whatever it was designed to do, and the sandbox records every action file system changes, network connections, process creation, registry modifications while the rest of your network remains completely unaffected.

Sandboxes are especially valuable for detecting zero-day exploits and novel malware that signature-based antivirus has never seen before. A file might have no detection on any antivirus engine, but when it runs in a sandbox and immediately starts making outbound connections to a command-and-control server or modifying the registry that behavior tells you everything you need to know.

**There are two types of sandbox deployment:**

**Local sandboxing:** Integrated directly into your email gateway or security appliance. Attachments are submitted automatically when they arrive and analyzed before delivery. This gives you low-latency, immediate results. Cisco Email Security supports this with integration to Cisco Threat Grid.

**External sandboxing:** Files are submitted to a cloud-based sandbox platform for deeper analysis. This allows for more complex simulations different operating systems, different user behavior patterns, longer observation windows. Use this for deeper investigation of files you have already quarantined.

> **Note:** sophisticated malware can detect that it is running inside a sandbox and will behave differently or not execute at all until it believes it is on a real system. This is called sandbox evasion. It is a real problem, and it is why manual malware analysis remains an important skill even with automated sandboxes available. Some of the most advanced threats will only show their true behavior when they sense they are in a real production environment.

## 5.6 Defensive Controls Against Malicious Attachments

Defending against attachment-based attacks requires multiple layers. No single control is sufficient on its own.

**File type whitelisting:** Define which file types are allowed to arrive as email attachments. Block executable files (.exe, .bat, .cmd, .ps1, .vbs) by default. Allow only what your business actually needs Word documents, PDFs, images. Anything not on the whitelist is blocked or quarantined automatically. In Cisco Email Security, this is configured under Content Filters using the Attachment Type condition.

**Macro detection:** Configure your email gateway to detect Office documents containing macros and quarantine them for review. Cisco Email Security has a Macro Detection option under its content filter conditions. When an Office file with embedded macros is detected, it is held rather than delivered. The security team reviews it before release.

**Attachment sandboxing:** Route all attachments through a sandbox before delivery. Any file that exhibits suspicious behavior in the sandbox is quarantined. This catches malware that bypasses signature detection.

**Content Disarm and Reconstruction (CDR):** CDR removes all potentially executable elements from a document macros, embedded scripts, active content and reconstructs a clean version of the file. The user receives a safe version of the document that still contains the readable content but has had all weaponizable components stripped out. This is covered in the next session.

**Use a different antivirus on your email gateway than on your endpoints:** This is a best practice that is often overlooked. If you run McAfee on your endpoints, do not run McAfee on your email gateway. Use a different engine Sophos, CrowdStrike, or another vendor. The reason: each antivirus engine

has its own signature database and detection logic. A file that slips past one engine will more likely be caught by a different vendor's engine. Defense in depth applies to antivirus too.

## 5.7 Simulating Attachment Attacks GoPhish

GoPhish is an open-source phishing simulation framework. As a Purple Team or Blue Team member, you use it to run internal phishing campaigns — sending test emails with simulated malicious attachments to your own users to measure their awareness and test your detection controls.

GoPhish lets you create realistic phishing templates — email layout, sender name, subject line — and attach files or include links. When a user clicks the link or opens the attachment, GoPhish records it. After the campaign, you get a report showing exactly who clicked, when, and from where. This data helps you identify which departments or individuals need more security awareness training, and it also tests whether your email gateway is catching these simulated attacks before they reach users.

## 5.8 User Awareness The Last Line of Defense

Technical controls can fail. A password-protected dropper bypasses gateway scanning. A zero-day exploit bypasses antivirus. When the technical layer fails, the user is the last line of defense. And in most attacks, the user is also the weakest link.

**The minimum level of awareness every user in your organization should have:**

**If you receive an email you were not expecting** even from someone you know do not open attachments or click links without verifying first. Contact the sender through a separate channel (Teams, phone, in person) to confirm they actually sent it.

**Always check the From domain** not just the display name. An email can show "Amazon Support" in the display name but come from a completely different domain. Look at the actual email address, not just the name.

**Never click "Enable Content" or "Enable Macros" in a document** unless you specifically requested a file that requires it and you know exactly why macros are needed. When in doubt do not enable it.

Awareness alone will not stop every attack that is not a realistic expectation. But awareness combined with technical controls makes your organization dramatically more resilient. Neither layer is sufficient alone. Both are necessary.