



SPF, Explained

How your domain tells the world
which computers may send email in your name

A plain-English guide for people who run businesses, not mail servers.
No prior knowledge assumed. Everything here can be checked without
touching a command line.

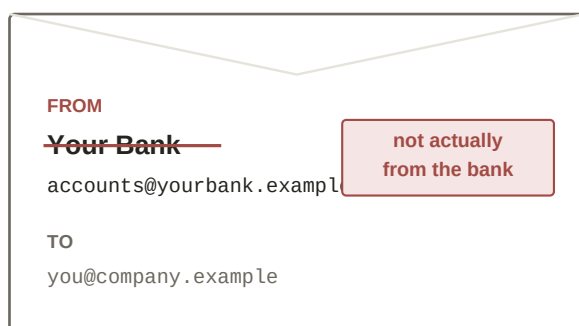
01

The problem SPF solves

Email was designed by people who trusted each other, and it shows. Anyone can claim to be anyone.

When a letter arrives, the return address on the envelope is simply whatever the sender chose to write. Nobody checked it. You already know this about post, and you adjust for it without thinking.

Email works exactly the same way. The name and address a message claims to come from are typed in by whoever sent it. A stranger can put your company in that line as easily as you can, and by default nothing anywhere checks whether they were entitled to.



Nothing stops this.

The post office never checks whether the return address on an envelope is true. It was never designed to. Email was built the same way, for the same reason: in 1982 everyone on the network was trusted.

SPF is the missing check.

A way for the recipient to ask: is this really you?

Figure 1 — A return address is a claim, not a fact. Email inherited this exactly.

This is why invoices appear to come from your finance team, and why a supplier receives a payment-details change that you never sent. The attacker does not need to break into anything. They just write your name on the envelope.

What SPF actually is

A short, public note attached to your domain name that lists which computers are allowed to send email on your behalf. Anyone receiving a message that claims to be from you can read that list and check. That is the whole idea. Everything else in this document is detail.

What happens when a message arrives

Five steps, all automatic, all finished in well under a second.



Figure 2 — The receiving server pauses mid-delivery to read your public list, then decides.

The important part is who does the work. **You publish the list once.** The checking is done by every receiving server, every time, without you being involved and without the sender being able to interfere. That is what makes it trustworthy: the answer comes from your domain, not from the message.

Where the list actually lives

In DNS — the same public directory that turns `yourdomain.com` into an address a browser can reach. Your domain registrar or hosting provider gives you a form to add entries. An SPF record is one of those entries, of type **TXT**, and it looks like this:

```
v=spf1 ip4:203.0.113.7 include:_spf.google.com ~all
```

One line. That is a complete, working SPF record, and the next section takes it apart piece by piece.

The most important thing to understand

There are two "from" addresses on every email, and SPF only checks the one nobody sees.

This catches out almost everyone, including people who work in IT. Every email carries two separate sender addresses, and they do not have to match.

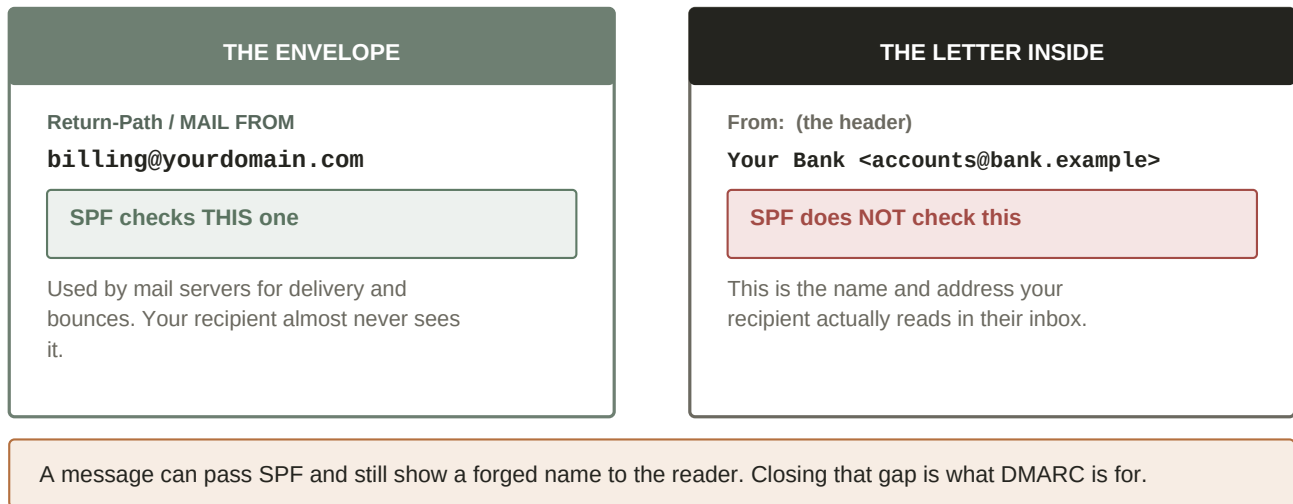


Figure 3 — SPF validates the envelope. Your recipient reads the letter inside.

A forger can therefore pass SPF perfectly honestly — using their own domain on the envelope, which is genuinely theirs — while writing *your* company in the From line that your customer actually reads. SPF passes. The message is still a forgery.

So is SPF pointless?

No — but on its own it is incomplete, and anyone who tells you SPF stops impersonation is wrong. SPF is one of three parts. It is the simplest and the right one to do first, and Section 8 explains how the other two close this gap.

Reading an SPF record

Four kinds of piece, in a fixed order, ending in a rule about everyone else.

v=spf1 ip4:203.0.113.7 include:_spf.google.com ~all

v=spf1

The version. Every SPF record starts with exactly this, and a record without it is ignored entirely.

ip4:203.0.113.7

A specific computer, named by its address. Use this for a server you run yourself.

include:_spf.google.com

Trust whatever list this other company publishes. This is how you authorise Google, Microsoft, Mailchimp and the rest without tracking their addresses yourself.

~all

The catch-all ending: what to do about everyone NOT listed above. This one says "treat as suspicious".

Order matters: the first mechanism that matches wins, so the catch-all always goes last.

Figure 4 — The same record, whole and then in pieces.

The pieces you will meet

Piece	What it means	When you would use it
include:	Trust the list published by someone else	Almost always. This is how you authorise Google Workspace, Microsoft 365, Mailchimp and similar.
ip4: / ip6:	One specific computer, named by address	A server you run yourself, or one your provider tells you to add.
a	The computer your website runs on	Only if that machine genuinely sends mail. Often added by mistake.
mx	The computers that receive your mail	Only if they also send. Receiving and sending are not the same job.
all	Everyone not already matched above	Always, exactly once, at the very end.

One record, one domain

A domain must have exactly one SPF record. Two records is not twice the protection — it is a syntax error, and well-behaved receivers treat the result as unusable. If you add a second sender, extend the existing line rather than adding a new one.

The ending is the part that matters

Everything before the last few characters is a list. The last few characters decide whether anyone acts on it.



Figure 5 — What to do about mail from anywhere else.

A list with no instruction attached is just information. The ending is the instruction. Publishing a beautifully accurate record that finishes with ?all achieves close to nothing, because you have told receivers you have no opinion.

A sensible order of operations

Start at ~all. Watch for a few weeks and find out who is actually sending in your name — there is almost always someone you had forgotten, a payroll system or a booking tool. Once nothing legitimate is being caught, move to -all. Going straight to -all on a domain you have not measured is how a company stops receiving its own invoices.

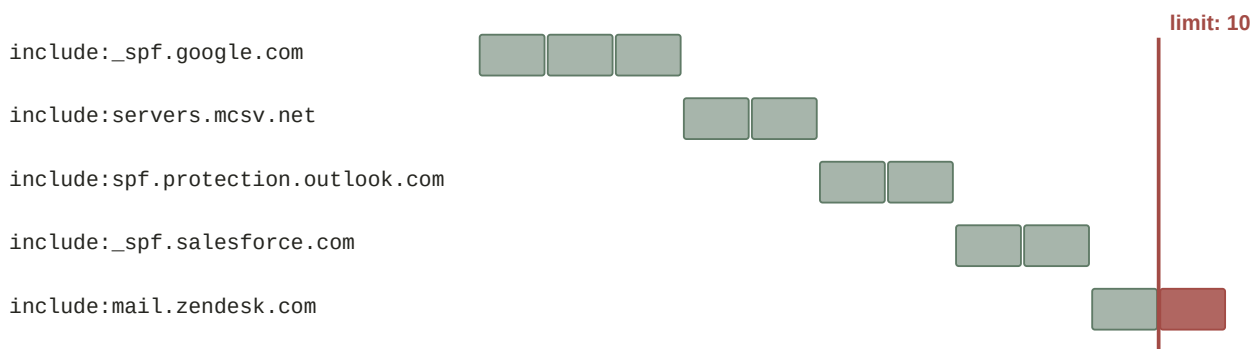
The limit that breaks records quietly

SPF allows ten lookups. Exceeding it does not degrade gracefully — it fails.

Every `include:` means the receiving server has to go and fetch another list. Those fetches are capped at ten, and the cap counts everything: an include you add may itself contain includes, each one spending from the same budget.

Includes are the usual culprit, but they are not the only things that cost. `a`, `mx`, `ptr`, `exists` and `redirect` each spend one too. Only `ip4:` and `ip6:` are free, because an address needs no looking up.

Each `include:` costs one lookup. Some cost several, because the list you include has includes of its own.



Past ten, the check does not just stop counting — it fails.

Mail from the providers listed after the tenth lookup stops authenticating, and nothing tells you.

Figure 6 — Five providers, eleven lookups. The fifth stops authenticating.

The reason this is dangerous rather than merely annoying is that nothing visible changes. The record still looks correct. Mail still sends. It simply stops being trusted, gradually, as providers are added over the years by people who each added only one.

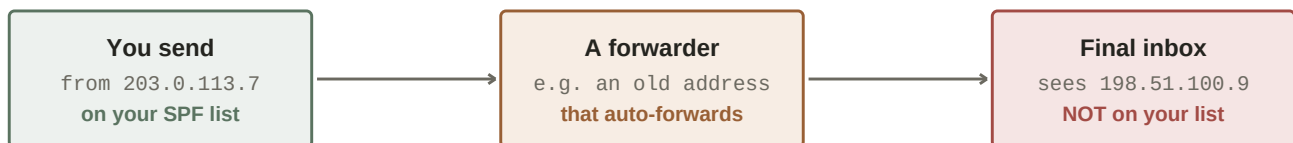
How to check

Run your domain through an SPF checker that reports the lookup count — ours is at reapercloud.com/dmarc.php. If the number is eight or higher, treat it as a problem to schedule rather than a crisis: remove providers you no longer use, and ask remaining providers whether they publish a narrower record you can use instead.

What SPF will not do for you

Four honest limitations. None of them are reasons to skip SPF; all of them are reasons not to stop there.

It breaks when mail is forwarded



The message is genuine. The forwarder is honest. SPF still fails.

SPF checks the computer that made the final connection, and after a forward that computer belongs to the forwarder, not to you. This is a known limitation, not a misconfiguration — and it is the main reason DKIM exists alongside SPF.

Figure 7 — A genuine message, an honest forwarder, and a failed check.

It does not protect the name your reader sees

Covered in Section 3, and worth repeating because it is the gap most often assumed to be closed. SPF checks the envelope; your customer reads the letter.

It is not encryption

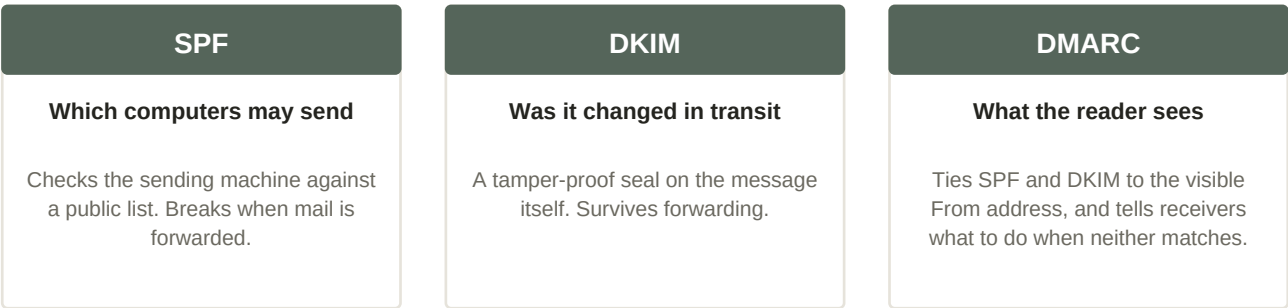
SPF says nothing about whether a message was readable in transit. It is about identity, not privacy. Those are separate problems with separate answers.

It does not stop lookalike domains

A perfect SPF record on `yourcompany.com` does nothing about someone registering `yourcompany-invoices.com` and publishing a perfect SPF record of their own. Their mail will authenticate beautifully. It is still fraud.

SPF, DKIM and DMARC

Three parts that only do their job together.



Why you need more than SPF

SPF alone does not stop someone putting your company name in the From line. It was never designed to. Publish SPF first because it is the simplest of the three, then DKIM, then DMARC to bind them to the address your customers actually read.

Figure 8 — SPF and DKIM each answer a narrow question. DMARC connects them to what the reader sees.

The order to adopt them is the order of difficulty. SPF is one DNS entry and an afternoon of finding out who sends on your behalf. DKIM is usually a switch your mail provider already offers. DMARC is the one that needs watching before it is tightened, because it is the one with teeth.

If you do only this	You get
Nothing	Anyone can send mail as your domain, and receivers have no way to tell.
SPF	Receivers can tell whether the sending computer was authorised — but not whether the name shown to the reader is yours.
SPF + DKIM	The above, plus proof the message was not altered, which survives forwarding.
SPF + DKIM + DMARC	The name your customers actually see is protected, and you receive reports showing who is sending as you.

A short checklist

What to ask for, and how to tell whether it was done.

Getting it right

- **List every service that sends mail as you.** Not just your email provider — invoicing, CRM, booking systems, the website contact form, payroll, the ticketing system. This list is always longer than expected and is the real work.
- **One SPF record, not several.** Extend the existing line when you add a provider.
- **Start at ~all, finish at -all.** Measure in between.
- **Stay under ten lookups.** Check the count whenever you add a provider, not once a year.
- **Use each provider's published include.** Do not copy their IP addresses by hand — they change, and yours will not.

Signs something is wrong

- Your record ends in +all. This authorises the entire internet to send as you, and is worse than having no record at all.
- Your record contains pt r. It is deprecated, slow, and unreliable; remove it.
- There are two SPF records on the domain. Merge them.
- The lookup count is at or above ten. Some of your mail is already failing.
- Nobody can say which services send mail as your company. Start there rather than with the record.

Check your own domain in about a minute

ReaperCloud publishes free tools that read your record the way a receiving server does, count the lookups, and explain any finding in the same plain language as this document. Start at reapercloud.com/report.php for everything at once, or reapercloud.com/dmarc.php for SPF and DMARC specifically.

This document describes SPF as specified in RFC 7208. It is an explanation, not a specification: where the two differ, the RFC is correct and this is simplified.