The Bottom-Most Received Header

Deterministic Sender Verification via Unforgeable SMTP Network Metadata

How the TCP/IP Layer Provides Ground-Truth Email Origin Data That Policy-Based Frameworks Cannot Match

Classification: Technical Whitepaper | **Domain:** Email Security Research | **Year:** 2026 **Audience:** Security Engineers, Email Administrators, CISOs, Security Researchers **Standards Referenced:** RFC 5321 • RFC 5322 • RFC 7208 • RFC 6376 • RFC 7489 • CVE-2023-51764

Executive Summary

Every email message that traverses the public internet carries a chain of `Received` headers — trace fields prepended by each Mail Transfer Agent (MTA) that handles the message in transit. Among these, the bottom-most `Received` header occupies a uniquely privileged position: it is inserted by the first server *outside the sender's control* — the first receiving MTA at the boundary of the recipient's network — and it records the TCP source IP address of the connection it accepted. This IP address is an OS-level network observation, not a protocol field the sending client can populate, override, or suppress. It is, accordingly, the single most forgery-resistant piece of metadata available in an email message, requiring no cryptographic infrastructure and no cooperation from the sending domain.

Policy-based authentication frameworks — Sender Policy Framework (SPF), DomainKeys Identified Mail (DKIM), and Domain-based Message Authentication, Reporting & Conformance (DMARC) — represent meaningful

advances in email security but are fundamentally contingent on correct deployment by the sending domain. As of Q2 2025, 61.9% of the top ten million domains publish no SPF record, 81.6% lack any DMARC record, and of those with DMARC, more than half enforce no policy whatsoever. These frameworks can be misconfigured, absent, bypassed via forwarding, or circumvented through documented protocol-level attacks such as SMTP Smuggling (CVE-2023-51764). The bottom-most `Received` header requires none of this cooperation: it is a structural property of TCP/IP that defending infrastructure can exploit unilaterally. This paper describes the mechanism, its evidentiary value, its limitations, and how email administrators, security operations teams, and vendors should integrate it into layered defenses.

Table of Contents

---

1. Introduction: The Email Identity Problem

Email phishing has reached industrial scale. Current estimates place the volume of phishing emails sent globally at between **3.4 and 4 billion messages per day**, accounting for approximately 1.2% of all global email traffic. Google's filtering infrastructure alone blocks more than 15 billion unsolicited or malicious messages daily. The Anti-Phishing Working Group (APWG) recorded 1,003,924 distinct phishing attacks in Q1 2025 alone — the largest quarterly total since late 2023 — and the overall trajectory has been sharply upward, with phishing attacks rising 182% since 2021. Human error, often induced by convincingly spoofed email, contributes to approximately 60% of all data breaches according to the Verizon 2025 Data Breach Investigations Report. The financial consequences are severe: Business Email Compromise (BEC) alone was responsible for $2.8 billion in reported losses in the United States in 2024.

At the root of this crisis is a design decision made half a century ago. The Simple Mail Transfer Protocol, first codified in RFC 821 (1982) and modernized through RFC 5321 (2008), was architected in an era of cooperative academic networks where trust between participants was assumed rather than enforced. SMTP's `From` header — the field that email clients prominently display as the sender's identity — is an arbitrary, user-supplied string. The protocol imposes no mechanism to verify that the address in the `From` field corresponds to the actual originating infrastructure. An attacker can set the `From` field to `ceo@fortune500company.com`, `security@paypal.com`, or any other identity, and the SMTP protocol will faithfully deliver the message without challenge.

Three decades of retrospective engineering — SPF (RFC 7208), DKIM (RFC 6376), and DMARC (RFC 7489) — have produced meaningful but incomplete countermeasures. Each framework requires proactive, correct deployment by the domain owner. Each has documented failure modes, bypass techniques, and adoption gaps measured in the hundreds of millions of domains. The security posture of any given recipient therefore depends critically on decisions made by parties entirely outside the recipient's control.

This paper advances a complementary thesis: the SMTP protocol, as currently deployed, already contains an unforgeable record of the message's true network origin — embedded not by the sender, but by the receiving infrastructure itself. This record, the **bottom-most `Received` header**, carries the TCP source IP address of the SMTP connection as observed at the OS network layer of the first receiving server. No SMTP command, no header manipulation, no DNS configuration can alter this value after the connection is accepted. Defenders can read it unilaterally. Its evidentiary value is deterministic rather than probabilistic. Understanding and systematically exploiting this signal represents an underutilized, zero-cooperation mechanism for sender origin verification.

Core Thesis

The bottom-most `Received` header in an email message contains the originating IP address as recorded by the first trusted receiving server. Because this IP is observed at the TCP/IP socket layer — not reported by the sender — it cannot be forged, suppressed, or altered by the email originator. It provides cryptography-free, policy-free, deterministic ground truth about message origin.

---

2. SMTP End-to-End Delivery: How Email Actually Travels

To understand why the bottom-most `Received` header is authoritative, it is necessary to trace the full lifecycle of an email message from composition to delivery, and to understand precisely when and how each `Received` header is inserted.

2.1 The SMTP Delivery Pipeline

A modern email message passes through a layered series of components, each with a defined role in the RFC 5321 architecture:

11. **Mail User Agent (MUA):** The end-user client — Outlook, Thunderbird, Apple Mail, a web interface — where the message is composed. The MUA constructs the message headers (including `From`, `To`, `Subject`, `Date`) and the body. All of these fields are entirely under sender control at this stage.
12. **Mail Submission Agent (MSA):** The sender's outbound mail server, typically listening on TCP port 587 with authenticated submission (RFC 4409). The MUA authenticates to the MSA, which accepts and queues the message for delivery. The MSA may perform outbound policy checks and will typically add the first `Received` header for internal audit purposes.
13. **Mail Transfer Agent (MTA) — Sending Side:** The MSA hands the message to an outbound MTA (or acts as one), which performs DNS MX record resolution for the recipient domain and establishes an SMTP TCP connection to the recipient's designated mail server. The sending MTA presents its identity via the `EHLO` command — a self-reported, unverified hostname.
14. **Intermediate Relay Hops:** Depending on organizational architecture, the message may traverse additional relay servers — outbound security gateways, Email Service Providers (ESPs), compliance archiving services — each adding its own `Received` header as it processes and forwards the message.
15. **Destination MX Server:** The recipient domain's inbound mail server, identified via DNS MX record lookup. This is the first server outside the sender's infrastructure. When it accepts the TCP SMTP connection, it records the source IP of that connection and prepends a `Received` header containing that IP. This is the bottom-most Received header from the perspective of the recipient's trust boundary.
16. **Final Delivery:** The destination MX server delivers the message to the recipient's mailbox via LMTP or local delivery — or relays it to an internal server, adding another `Received` header above the previous ones.

2.2 DNS MX Record Resolution

When the sending MTA is ready to deliver a message, it queries the DNS for `MX` (Mail Exchanger) resource records associated with the recipient's domain. For example, a message addressed to `user@example.com` triggers a DNS query for `MX example.com`. The DNS response returns one or more hostnames with associated priority values. The sending MTA attempts delivery to the highest-priority (lowest numeric value) MX host first, falling back to lower-priority hosts on failure. The address the sending MTA connects to is determined entirely by this DNS response — the sender cannot dictate which MX server it reaches.

2.3 The Received Header Chain: Chronological Structure

RFC 5321, Section 4.4 is unambiguous: *"When an SMTP server receives a message for delivery or further processing, it MUST insert trace ('time-stamp' or 'Received') information at the beginning of the message header."* Each MTA in the chain prepends its own `Received` header, meaning the message accumulates these headers in reverse chronological order:

- The **topmost** `Received` header is the most recently added — typically the final internal delivery hop.
- The **bottom-most** `Received` header is the oldest — added by the first MTA to receive the message *outside the sender's infrastructure*.

This stack-like structure means that reading from the bottom up traces the message's path from origin to inbox.

2.4 Anatomy of a Received Header

A fully annotated `Received` header illustrates each field's evidentiary weight:

```
Received: from mail-sender-hostname.evil.net           # EHLO/HELO name —
self-reported, unverified           (mail-sender-hostname.evil.net
[203.0.113.47])  # Parenthetical: server-observed TCP source IP           by
mx1.recipient.com                           # Receiving server's own hostname
with ESMTPS                                    # Protocol: Extended SMTP over
TLS           id a1b2c3d4e5f6                                    # Internal
message ID assigned by receiver           for <user@recipient.com>;
# Envelope recipient           Tue, 26 May 2026 20:47:31 +0000 (UTC)
# Timestamp added by receiving server
```

The critical field is the parenthetical IP address — here **203.0.113.47**. This value was not transmitted by the sender in any SMTP command. It was recorded by the

receiving server (`mx1.recipient.com`) from the TCP socket's source address at the OS network layer. The sender cannot set, alter, or suppress this value. Every other field in this header — including the EHLO hostname immediately preceding the parenthetical — is sender-reported and therefore untrusted.

> **Key Insight**
>
> The receiving server's parenthetical IP observation is an OS-level network fact. No SMTP command exists to alter the source address of a TCP socket after the connection is established. The sending client did not write this field — the receiving server did, from kernel socket metadata.

2.5 The "First Server Outside Sender Control" Principle

It is important to be precise about what "bottom-most" means in a trust context. A sophisticated attacker could craft a message that already contains fabricated `Received` headers — injected as part of the message body before submission — in an attempt to simulate a multi-hop path and confuse analysis. These fabricated headers, however, will always appear *below* the genuine `Received` header added by the first server that actually accepted the TCP connection from the sender. The receiving MTA cannot be instructed to place its header below pre-existing content; it prepends unconditionally, per RFC 5321 §4.4. Accordingly, the true bottom-most legitimate `Received` header is the one added by the first MTA the recipient controls — and distinguishing it from injected fakes is straightforward: it will be the lowest in the stack that was not present when the sending MTA initiated the connection.

---

3. The Unforgeable Nature of TCP/IP Network Metadata

The assertion that the parenthetical IP in a `Received` header is unforgeable deserves rigorous justification, because it is the load-bearing claim of this paper's thesis.

3.1 The TCP Three-Way Handshake as a Trust Anchor

SMTP operates over TCP, a connection-oriented transport protocol. Before any SMTP dialog can occur, the client and server must complete the TCP three-way handshake:

17. **SYN:** The client sends a TCP SYN packet from its source IP address to the server's IP on port 25 (or 587/465).
18. **SYN-ACK:** The server responds with a SYN-ACK packet sent to the client's source IP address. If the source IP is fabricated, this packet travels to the fabricated address — which the attacker does not control.
19. **ACK:** The client must send an ACK back, completing the handshake. Only then does SMTP data exchange begin.

For an attacker to forge the source IP at the TCP layer, they would need to receive the server's SYN-ACK packet at the spoofed address (which they do not control), correctly predict the TCP sequence number, and send an appropriately crafted ACK — all without being able to see the server's response. Modern TCP implementations use cryptographically random initial sequence numbers (RFC 6528) specifically to defeat such blind TCP injection attacks. Even in environments where ISP egress filtering is absent, TCP-level IP spoofing is not a viable attack vector for establishing a full-duplex SMTP session.

3.2 The Distinction Between Forgeable and Unforgeable Fields

It is essential to distinguish between two categories of data in an email message:

Field Category	Examples	Who Writes It	Forgeable?
Message Header Fields	`From`, `Reply-To`, `Subject`, `Date`, `MIME-Version`	Sender (MUA)	Yes — arbitrary string
SMTP Envelope	`MAIL FROM`, `RCPT TO`	Sender (MTA)	Yes — sender-supplied
EHLO / HELO Hostname	The hostname string in `EHLO mail.example.com`	Sender (MTA)	Yes — self-reported
Received	`from` clause, `by`	Receiving MTA	Partially —

Field Category	Examples	Who Writes It	Forgeable?
Header (text portion)	clause, timestamp		timestamp and `by` are authoritative; `from` reflects sender's EHLO
Received Header (parenthetical IP)	`[203.0.113.47]` in `(hostname [203.0.113.47])`	Receiving MTA (from OS)	**No — OS socket observation**

3.3 Injected Received Headers vs. Authentic Ones

A frequently raised objection is that a sender could prepend fake `Received` headers to their message before transmission, fabricating an elaborate routing trail. This is correct — and it is precisely why the *position* of the header in the stack matters. Consider the following invariant: when the first recipient-controlled MTA accepts the SMTP connection, it calls `prepend_header("Received", ...)` using kernel socket metadata. This action places the authentic header at the top of whatever stack the message arrived with. The receiver's `Received` header is therefore always above any headers the sender injected. Reading the stack from the bottom upward, the first `Received` header whose `by` clause names a server the recipient controls is the authentic boundary marker.

3.4 The Relay and ESP Attack Surface

When a message transits a legitimate Email Service Provider (ESP), cloud sending platform, or forwarding service, the bottom-most `Received` header at the recipient's boundary will reflect the ESP's IP address rather than the ultimate original sender. This is not a failure of the mechanism; it is informative. The ESP's identity is disclosed by the IP and its PTR record. If the ESP's IP is not consistent with legitimate use of that ESP by the claimed sender domain, this itself is a strong phishing signal. Furthermore, the full chain of `Received` headers can be traversed upward to identify the IP that submitted the message to the ESP — often recoverable from the ESP's own injected trace headers.

RFC Reference

RFC 5321, Section 4.4 mandates that receiving servers MUST insert `Received` headers. RFC 5322, Section 3.6.7 defines the format of trace fields in the Internet Message Format. Neither standard provides any mechanism for the sending client to control, suppress, or modify the receiving server's `Received` header entry.

---

4. Reading the Chain: Locating the True Origin

4.1 Step-by-Step Methodology

The following procedure, applicable to any raw email message, extracts the authoritative network origin information:

20. **Access raw headers.** In most email clients, raw headers are accessible via "Show Original," "View Source," or equivalent options. The raw view presents headers in their original, unrendered form, including all `Received` headers that the client's rendering layer may suppress.
21. **Identify all `Received` headers.** Each `Received:` line in the header block represents one hop in the delivery chain. There may be two, three, five, or more, depending on routing complexity.
22. **Navigate to the bottom-most `Received` header.** In raw header order, this is the last `Received` header in the file — the one appearing furthest toward the body of the message. It is chronologically the oldest: the first hop recorded.
23. **Extract the parenthetical IP address.** Within the bottom-most `Received` header, locate the parenthetical expression following the `from` clause. It takes the form `(ehlo-hostname [IP.ADDRESS])`. The bracketed IP address is the originating TCP source IP.

24. **Perform a reverse DNS (PTR) lookup.** Query the DNS for the PTR record of the extracted IP address. This returns the canonical hostname registered by the IP's network operator — typically the sending mail server's official fully-qualified domain name (FQDN).
25. **Cross-reference against the claimed sender domain.** Compare the PTR-resolved hostname's apex domain against the domain in the `From` header. A legitimate email from `support@paypal.com` should show a PTR record resolving to a hostname within `paypal.com`, `paypal-corp.com`, or a known ESP operating on PayPal's behalf — verifiable against PayPal's published MX and A records.

4.2 Interpreting Matches and Mismatches

Match: The PTR-resolved domain of the bottom-most Received IP aligns with the From domain (or a known, documented sending infrastructure of that domain). This is a strong positive signal for legitimacy. It does not conclusively prove the message is benign — a compromised legitimate account would also match — but it eliminates the largest class of opportunistic phishing.

Mismatch: The PTR-resolved domain does not correspond to the claimed From domain. This is a high-confidence phishing indicator. The message may still be legitimate if the sender routes through an undocumented ESP or relay, but unexplained mismatches warrant immediate investigation.

4.3 Annotated Example A: Legitimate Email

```
Return-Path: <statement-noreply@legitimatebank.com> Received: from
mail.company-internal.com (mail.company-internal.com [10.0.1.5])           by
mx2.company-internal.com with ESMTP id xyz123           Tue, 26 May 2026
20:47:45 +0000           # Internal delivery hop — disregard for origin
analysis  Received: from filter01.company-internal.com (filter01.company-
internal.com [10.0.1.8])           by mail.company-internal.com with ESMTP id
abc456           Tue, 26 May 2026 20:47:42 +0000           # Internal
security gateway — also internal  Received: from mail-out4.legitimatebank.com
(mail-out4.legitimatebank.com [198.51.100.84])           by mx1.company.com
with ESMTPS id def789           Tue, 26 May 2026 20:47:38 +0000           #
Bottom-most Received — this is the FIRST server outside sender's control
# IP: 198.51.100.84           # PTR lookup on 198.51.100.84 → mail-
out4.legitimatebank.com           # Apex domain: legitimatebank.com ✓ Matches
From domain  From: "Legitimate Bank Alerts" <statement-
noreply@legitimatebank.com> Subject: Your March 2026 Statement Is Ready
```

Verdict: PTR record resolves to `legitimatebank.com`. The originating IP is consistent with the claimed sender domain. High confidence of legitimacy.

4.4 Annotated Example B: Phishing Email

```
Return-Path: <bounce@legitimatebank-secure-alerts.com> # Fabricated received
headers added by attacker to confuse analysis: Received: from
mail1.legitimatebank.com (mail1.legitimatebank.com [198.51.100.10])
by relay.legitimatebank.com with ESMTPS id fakeA           Tue, 26 May 2026
20:32:11 +0000           # INJECTED BY ATTACKER — NOT authentic  Received:
from relay.legitimatebank.com (relay.legitimatebank.com [198.51.100.1])
by inbound.legitimatebank.com with ESMTPS id fakeB           Tue, 26 May 2026
20:31:50 +0000           # INJECTED BY ATTACKER — NOT authentic  Received:
from vps-node-47.ru-host.example.net           (vps-node-47.ru-
host.example.net [45.142.212.33])           by mx1.company.com with ESMTPS id
realHop           Tue, 26 May 2026 20:31:30 +0000           # REAL bottom-
most Received — added by OUR mx1 server           # IP: 45.142.212.33
# PTR lookup → vps-node-47.ru-host.example.net           # Apex domain: ru-
host.example.net ✗ Does NOT match "legitimatebank.com"           # ASN:
AS206728 — Eastern European VPS provider. AbuseIPDB: 94% confidence malicious
From: "Legitimate Bank Security" <security@legitimatebank.com> Subject:
URGENT: Your Account Has Been Suspended
```

Verdict: Despite two injected `Received` headers attempting to simulate origin from `legitimatebank.com`, our own MTA's bottom-most `Received` entry exposes the true origin: a VPS in an unrelated hosting network. The injected headers are above the authentic one in raw format — reading the stack correctly reveals the deception immediately.

---

5. Comparison with SPF, DKIM, and DMARC

5.1 SPF (Sender Policy Framework) — RFC 7208

SPF enables a domain owner to publish, in their DNS zone, a TXT record listing the IP addresses and hostnames authorized to send mail on behalf of that domain. A receiving MTA checks whether the connecting IP appears in the sending domain's SPF record during the SMTP transaction, before the message content is accepted. On match, SPF "passes"; on failure, the configured result (`~all` softfail, `-all` hardfail, `?all` neutral, or `+all` pass-all) is applied.

SPF has several well-documented structural weaknesses:

- **Envelope vs. header mismatch:** SPF authenticates the *envelope sender* (the `MAIL FROM` Return-Path address), not the visible `From` header. An attacker

can pass SPF with a legitimate bounce address while spoofing the displayed `From` field entirely.

- **Forwarding failures:** When a message is forwarded by an intermediate server, the source IP changes but the envelope sender does not. SPF checks now fail for legitimate forwarded mail, a fundamental protocol incompatibility.
- **DNS lookup limit misconfiguration:** RFC 7208 limits SPF evaluation to 10 DNS lookups. Complex organizations with many `include:` mechanisms routinely exceed this limit, causing SPF to return `permerror` — which many receivers treat as a pass to avoid blocking legitimate mail.
- **Dangling include domains:** SPF records that include third-party domains (`include:esp.example.com`) become permanently open if that third-party domain lapses or is re-registered by an attacker.
- **The `+all` catastrophe:** Approximately 0.045% of SPF-publishing domains use `+all`, which authorizes any IP in the world to send as that domain — completely inverting SPF's purpose.
- **Adoption gap:** As of Q2 2025, 61.9% of the top ten million internet domains publish no valid SPF record.

5.2 DKIM (DomainKeys Identified Mail) — RFC 6376

DKIM allows a sending domain to cryptographically sign selected header fields and the message body using a private RSA or Ed25519 key. The corresponding public key is published in the domain's DNS as a TXT record at a designated selector subdomain. A receiving MTA retrieves the public key and validates the signature, confirming that the signed content was not altered in transit and that the signer controlled the corresponding private key.

DKIM's design assumptions create the following vulnerabilities:

- **DKIM replay attacks:** A valid DKIM signature persists indefinitely after signing. An attacker who receives a legitimately DKIM-signed email (e.g., a newsletter or transactional email) can redistribute it to thousands of

recipients. All copies will pass DKIM validation, because the signature is over the original message content — which has not been altered.

- **Weak key lengths:** Sub-1024-bit RSA DKIM keys have been demonstrated to be factorizable with commodity hardware. Studies have found significant proportions of published DKIM keys with insufficient key lengths.
- **The `d=` tag vs. the `From` header:** DKIM validates the signing domain (the `d=` tag), not the visible `From` header address. A message from `From: security@bank.com` can carry a valid DKIM signature with `d=attacker-esp.com`, and DKIM will pass. Only when DMARC alignment is enforced does this disparity matter.
- **Key rotation neglect:** DKIM key rotation is rarely practiced in most organizations, increasing the window of exposure for key compromise.
- **Compromised accounts:** A threat actor who compromises a legitimate sending account or obtains sending credentials from an ESP can generate validly DKIM-signed phishing messages through that infrastructure.

5.3 DMARC (Domain-based Message Authentication, Reporting & Conformance) — RFC 7489

DMARC builds upon SPF and DKIM by adding the critical concept of *identifier alignment*: requiring that at least one of SPF or DKIM pass using a domain that aligns with the visible `From` header address. DMARC also defines enforcement policy via the `p=` tag (`none`, `quarantine`, `reject`) and provides a reporting framework for senders to receive aggregate and forensic feedback.

DMARC is the most complete of the three policy-based frameworks, yet its real-world deployment reveals profound gaps:

- **Non-adoption:** 81.6% of the top ten million domains lack any DMARC record. Of the 18.2% with valid DMARC records, 10.6% use `p=none` — a monitoring-only policy with no enforcement effect. Only 3.9% enforce `p=reject` with subdomain coverage.

- **The one-or-the-other loophole:** DMARC requires that SPF *or* DKIM pass with alignment — not both. An attacker needs only to defeat one of the two mechanisms, not both simultaneously.
- **Mailing list and forwarding breakage:** Mailing lists commonly alter the message subject or add footer text, breaking DKIM signatures. Combined with forwarding-induced SPF failures, this frequently causes legitimate DMARC failures — pushing organizations toward `p=none` to avoid blocking legitimate mail.
- **SMTP Smuggling (CVE-2023-51764):** A 2023 vulnerability in Postfix (and related MTAs) demonstrated that inconsistent interpretation of SMTP end-of-data sequences between different server implementations allows attackers to inject spoofed messages that appear to originate from a legitimate Postfix server. These injected messages can pass SPF (because they appear to come from an authorized IP) and bypass DMARC enforcement entirely. While patched in recent versions, this class of attack illustrates that even correct DMARC implementation can be bypassed via protocol-level manipulation.
- **Sender dependency:** DMARC's effectiveness is entirely contingent on domain owners publishing correct records. The recipient's DMARC policy depends entirely on decisions made by billions of independent domain operators, most of whom lack the expertise or incentive to implement it correctly.

5.4 Comprehensive Comparison Table

Attribute	Bottom-Most Received Header	SPF (RFC 7208)	DKIM (RFC 6376)	DMARC (RFC 7489)
What It Authenticates	TCP source IP of the originating SMTP connection	Envelope sender (Return-Path) domain vs. authorized IPs	Signing domain (`d=` tag) and message content integrity	Alignment of SPF/DKIM domain with visible From header
Forgeable	**No** — OS-	**Yes** —	**Partial** —	**Partial** —

Attribute	Bottom-Most Received Header	SPF (RFC 7208)	DKIM (RFC 6376)	DMARC (RFC 7489)
by Sender?	level socket observation	envelope MAIL FROM is sender-controlled	signature is valid but From can differ from d=	requires only SPF or DKIM alignment, not both
Requires Sender DNS Config	**No** — receiver acts unilaterally	**Yes** — SPF TXT record required	**Yes** — public key DNS TXT record required	**Yes** — DMARC TXT record plus SPF and/or DKIM
Fails on Forwarding?	**Partially** — IP changes to forwarder; chain remains traceable	**Yes** — SPF fails when forwarder IP is not in sender's record	**Sometimes** — body modification breaks signature	**Often** — SPF and DKIM failures cascade to DMARC failure
Adoption Rate (top 10M domains)	100% — every SMTP connection generates this data	~36.7% valid records (Fortra Q2 2025)	Estimated ~55% of active sending domains	18.2% valid; only 3.9% enforce p=reject with subdomains
Known Bypass Techniques	Compromised sending account; shared hosting (IP reuse)	Forwarding, dangling includes, +all misconfiguration, SMTP Smuggling	DKIM replay, weak keys, compromised signing account, header injection	CVE-2023-51764 (SMTP Smuggling), mailing list rewrites, forwarding
Cryptographic Infrastructure Required	**None**	None (DNS only)	PKI (public/private key pair)	None (relies on SPF/DKIM infrastructure)
Verification Is Deterministic	**Yes** — binary IP fact	Mostly (permerror ambiguity)	Yes (but only for signing domain, not From)	Mostly (forwarding creates false negatives)

Important Distinction

SPF, DKIM, and DMARC are sender-side controls: their effectiveness depends entirely on what the sending domain chooses to publish and configure. The

bottom-most `Received` header is a receiver-side observation: it requires nothing from the sending domain and cannot be influenced by sender behavior, misconfiguration, or omission.

---

6. Why This Method Stops Phishing

The bottom-most `Received` header's power lies in a simple, immutable constraint: a sender cannot choose what IP address they appear to connect from. They are bound to their actual network infrastructure. Regardless of how convincingly an attacker fabricates the message's `From` field, subject line, HTML body, and even supporting DNS records, the TCP connection they use to deliver the message will always originate from an IP address they actually control — and that IP address will be recorded faithfully by the receiving server.

6.1 Phishing Attack Classes and Countermeasures

The following taxonomy covers the most prevalent phishing techniques and demonstrates how bottom-most Received header analysis defeats each:

a. Display Name Spoofing — The attacker sets the `From` header to `"Bank of America" <attacker@malicious-domain.ru>`. Many email clients prominently display only the display name, hiding the actual address. Bottom-most IP analysis immediately reveals that the originating IP belongs to infrastructure with no relationship to `bankofamerica.com`, regardless of the display name shown to the user.

b. Lookalike Domain Attacks — The attacker registers `bankofamer1ca.com`, configures SPF, DKIM, and DMARC for that domain, and sends phishing messages that pass all three policy-based checks. The bottom-most Received header will show an IP address belonging to `bankofamer1ca.com`'s sending infrastructure — which, while correctly authenticated as sending from that domain, is trivially distinguished

from `bankofamerica.com`'s known IP ranges. PTR lookup on the originating IP reveals the true sending domain, not the impersonated one.

c. Header From Spoofing with Legitimate Return-Path — The attacker uses a real domain for the envelope `MAIL FROM` (passing SPF) while setting the `From` header to an impersonated address. Without DMARC alignment enforcement, this attack passes SPF and potentially DKIM. Bottom-most Received analysis is immune to this distinction: it cares only about the originating TCP connection IP, not the envelope sender field.

d. Compromised Legitimate Accounts — An attacker who successfully compromises a legitimate email account and sends phishing from within that organization's genuine infrastructure will send from an IP address that legitimately belongs to that domain. In this case, the bottom-most Received header method agrees with SPF, DKIM, and DMARC — all four approaches will indicate a legitimate sending infrastructure. This is the attack class where network-level metadata cannot help, and where behavioral analytics, user behavior anomaly detection, and multi-factor authentication on mailbox access are the appropriate controls. The bottom-most Received header method does not claim to supersede all security controls; it excels at the cases that policy-based frameworks miss.

e. Relay-Based Attacks Through ESPs — Attackers who compromise or abuse ESP accounts send phishing through a legitimate ESP's infrastructure. The bottom-most Received header reveals the ESP's IP and PTR record. While the ESP itself is a legitimate entity, the sending organization's identity is disclosed. Cross-referencing the ESP's identity against the claimed From domain (e.g., `From: security@paypal.com` but originating from an ESP that PayPal does not use) provides a high-confidence phishing signal. Published ESP allowlists and BIMI indicators can assist in maintaining authoritative mappings of which ESPs are sanctioned by which domains.

6.2 Supplementary Network Signals

The originating IP extracted from the bottom-most `Received` header serves as a pivot point for several additional verification signals, each independently valuable:

- **PTR (rDNS) Record:** The primary verification signal — maps the IP to a canonical hostname, allowing domain comparison.
- **ASN Ownership:** WHOIS and BGP data identifies the autonomous system that owns the IP block, its country of registration, and its typical use classification (hosting provider, enterprise, residential ISP).
- **IP Geolocation:** Inconsistency between the claimed sender's geography and the IP's geolocation is a phishing indicator (e.g., a US-headquartered bank's "security alert" originating from an Eastern European data center).
- **Threat Intelligence Feeds:** Services such as AbuseIPDB, VirusTotal, and Shodan maintain scored reputation databases for IP addresses. An IP with prior abuse reports, Shodan-catalogued open services inconsistent with legitimate mail infrastructure, or VirusTotal detection history provides corroborating evidence of malicious use.

6.3 Determinism vs. Heuristics

A defining advantage of the bottom-most Received header method is its *deterministic* nature. Machine learning classifiers, behavioral analytics, and content-based heuristics operate probabilistically — they assign confidence scores, suffer from false positive rates, and can be gamed through adversarial optimization. The bottom-most Received header produces a binary, verifiable assertion: either the originating IP's PTR resolves to a domain consistent with the claimed sender, or it does not. This assertion can be encoded as an explicit, auditable rule in an email security gateway, a SIEM detection, or a mail filter — without any statistical uncertainty.

Key Insight

Unlike probabilistic heuristics or ML classifiers, the bottom-most Received header check produces a deterministic, binary result: the originating IP either belongs to infrastructure consistent with the claimed sender domain, or it does not. This makes it exceptionally well-suited for rule-based enforcement in email

security gateways.

---

7. Implementation Guidance

7.1 For Email Administrators

Email administrators are uniquely positioned to operationalize bottom-most Received header analysis at the infrastructure level, before messages reach end users:

- **Preserve and log full Received header chains.** Configure inbound MTAs (Postfix, Exim, Sendmail, Microsoft Exchange, Proofpoint, Mimecast) to log the complete Received header chain for every accepted message. Minimum retention of 90 days is recommended for forensic purposes. Ensure that any inline security gateways do not strip or modify Received headers from inbound messages.
- **Build origin IP extraction filters.** Implement milter-based (or equivalent) filters on inbound MTAs that:
    1. Parse the complete Received header stack at the point of acceptance.
    2. Identify the first externally-originated Received entry (i.e., the bottom-most header whose by clause names the organization's own MX).
    3. Extract the parenthetical IP address from that header.
    4. Perform a PTR lookup on that IP.
    5. Compare the PTR domain apex against the From header domain apex.
    6. Assign a mismatch flag to the message's internal scoring system.
- **Define mismatch policy thresholds.** Not all mismatches indicate phishing. Maintain an allowlist of known ESPs, forwarding services, and routing partners

whose IPs are expected to appear as the first external hop for legitimate mail. Apply heightened scrutiny to mismatches outside this allowlist, and consider quarantine or reject actions based on mismatch severity and additional signals.

- **Integrate with SIEM platforms.** Forward extracted origin IP, PTR domain, From domain, and mismatch flag to the organization's SIEM (Splunk, Microsoft Sentinel, Elastic SIEM). Correlate with threat intelligence enrichment to surface patterns across time and sender infrastructure.

7.2 For Security Operations

Security operations teams investigating phishing reports should adopt the following workflow, elevating the bottom-most Received header to the first step of any phishing triage:

- **Lead with header analysis.** When a user reports a suspicious email, the first analytical action should be extraction of raw headers and identification of the bottom-most Received IP. This single step often definitively confirms or refutes a phishing hypothesis within seconds.
- **Correlate the origin IP with threat intelligence.** Submit the extracted IP to AbuseIPDB (for abuse history), VirusTotal (for malware and phishing associations), and Shodan (for port/service profiling that may indicate bulletproof hosting or malicious infrastructure). An IP with a 90%+ AbuseIPDB confidence score for malicious activity is a near-definitive phishing confirmation.
- **Build standardized playbooks.** Formalize the following as a named step in all phishing investigation playbooks: "Step 1: Extract origin IP from bottom-most Received header. Perform PTR lookup. Compare to From domain. Submit to TI feeds." This ensures consistent application across analysts of varying experience levels.
- **Track origin IP infrastructure across campaigns.** Phishing campaigns frequently reuse IP infrastructure. Once a malicious IP is identified, pivot to historical message logs to identify all previously received messages from that

IP or ASN — revealing scope of a campaign and potential prior victim notifications.

7.3 For Developers and Vendors

Email security product vendors and developers building email processing applications should integrate this signal systematically:

- **Surface origin IP in security gateway UIs.** Email security gateways (Proofpoint, Mimecast, Abnormal Security, Defender for Office 365) should prominently display the bottom-most Received IP, its PTR record, and a visual indicator of alignment with the From domain on every message detail view. This information is currently buried in raw header views inaccessible to most analysts.
- **Expose origin IP as a programmatic API signal.** Email inspection APIs should return `origin_ip`, `origin_ptr`, `origin_domain`, and `ptr_from_alignment` (boolean) as first-class response fields alongside SPF, DKIM, and DMARC results. This enables downstream systems to incorporate the signal in their scoring logic.
- **Use as a supplemental, not replacement, signal.** Architect integrations that combine bottom-most Received header analysis with SPF, DKIM, DMARC, content analysis, and behavioral signals. No single control is sufficient. The bottom-most Received header is exceptionally strong for identifying infrastructure mismatch; it is not designed to detect account compromise or malicious content from legitimate infrastructure.
- **Implement header chain depth validation.** Build logic to detect anomalous numbers of `Received` headers (potential injection attack), headers with timestamps that are out of chronological order (indicating tampering), or `Received` headers whose `by` clauses claim to be the receiving organization's own servers (a telltale sign of injected fake headers).

Implementation Tip

When identifying the bottom-most "legitimate" Received header, filter out any Received headers whose `by` field names one of your own organization's MX servers or internal relay hosts. The first Received header (reading bottom-to-top) that was added by your own inbound MX boundary server is the authoritative one. Headers below it in the raw file were present when the message arrived — they were authored either by the sending infrastructure or injected by the attacker.

---

8. Limitations and Edge Cases

Intellectual honesty requires a clear accounting of the scenarios in which the bottom-most Received header method produces unreliable or ambiguous results. These limitations do not negate the method's value; they define the boundaries within which it should be applied.

8.1 Cloud Email Routing and Proxy Services

Organizations that route all outbound (and sometimes inbound) email through cloud security gateways — Microsoft Defender for Office 365, Google Workspace routing rules, Proofpoint/Mimecast inline gateways — may cause all inbound mail to arrive at the organization's MX boundary from the gateway vendor's IP ranges. In this architecture, the bottom-most Received header at the true boundary server reflects the gateway's IP, not the ultimate message originator. **Mitigation:** Maintain a documented allowlist of known cloud gateway IP ranges; trust their pre-inserted `X-Forefront-Antispam-Report`, `X-Google-DKIM-Signature`, or equivalent trace headers as proxies for the origin IP they recorded. Most enterprise email security gateways also inject their own proprietary trace headers containing the original sender IP.

8.2 IPv6 and Reverse DNS Gaps

IPv6 adoption complicates PTR-based verification significantly. Many legitimate organizations that have deployed IPv6 for outbound email sending have not correspondingly configured PTR (rDNS) records for their IPv6 address blocks, because PTR delegation for IPv6 requires explicit configuration with the ISP or block owner, and is frequently neglected. An IPv6 origin address with no PTR record is ambiguous — it may be a legitimate but poorly maintained sending infrastructure, or it may be a newly spun-up malicious VPS. **Mitigation:** Treat missing IPv6 PTR records as a moderate risk signal rather than a definitive mismatch indicator; supplement with ASN ownership analysis and cross-reference against the claimed sender domain's known IPv6 ranges.

8.3 Shared Hosting Environments

Shared hosting providers and multi-tenant cloud email platforms may route thousands of unrelated sending organizations through a small pool of IP addresses. In this scenario, the originating IP's PTR record reflects the hosting provider's infrastructure rather than the individual sender's domain. Legitimate email from `smallbusiness.com` hosted on a shared platform may originate from an IP whose PTR resolves to `mail-relay-12.sharedhost.com` — producing a PTR-to-From mismatch for a legitimate message. **Mitigation:** Maintain an allowlist of major shared hosting and cloud email providers whose IPs are known to legitimately host many different sender domains. Apply additional authentication signals (SPF, DKIM, DMARC) more heavily in this context.

8.4 Anonymization Networks and Proxy Chains

Sophisticated threat actors may route their SMTP connections through Tor exit nodes, commercial proxy services, or compromised servers acting as SMTP relays. In this case, the originating IP recorded in the bottom-most Received header reflects the exit node or compromised relay, not the ultimate attacker. The exit node's PTR record may even resolve to a superficially plausible hostname. **Mitigation:** Cross-reference the originating IP against known Tor exit node lists (updated daily from the Tor Project's consensus data), commercial proxy/VPN IP databases, and Shodan's service scans. ASN reputation — particularly for bulletproof hosting ASNs

— provides a valuable additional signal. AbuseIPDB maintains crowd-sourced abuse reports that frequently identify Tor exit nodes and proxy IPs.

8.5 VPNs Used by Legitimate Senders

Legitimate senders who submit email through their organization's mail infrastructure while connected via corporate VPN or consumer VPN services may cause their outbound traffic to appear from VPN exit nodes rather than their organization's declared infrastructure. This is a relatively rare scenario in enterprise contexts (where mail submission is typically handled by dedicated infrastructure, not the user's workstation), but may arise in smaller organizations or remote-work configurations. **Mitigation:** In enterprise email security contexts, user workstation IPs rarely appear as the bottom-most Received IP because submissions pass through authenticated MSAs; treat apparent consumer VPN IPs in this position as a risk signal worth investigating.

Important Caveat

The bottom-most Received header method is a high-confidence signal within a layered defense architecture — not a standalone replacement for SPF, DKIM, DMARC, content analysis, or threat intelligence. Its greatest strength is in the detection of opportunistic phishing and infrastructure impersonation. It should be applied with appropriate context, allowlists, and supplementary signals to minimize false positives in complex routing environments.

---

9. Conclusion

The bottom-most `Received` header in an SMTP email message is, in the strict sense of the term, an unforgeable artifact. It is placed there not by the sender, but by the receiving infrastructure — specifically, by the first MTA at the recipient's network boundary, recording the TCP source IP address of the SMTP connection it accepted.

This IP address is read from the kernel's socket metadata, not from any field the sending client can control. No SMTP command, no header injection, no DNS manipulation can alter it after the connection is established. It requires no sender cooperation, no cryptographic infrastructure, and no shared configuration: it is a structural property of TCP/IP and SMTP that defenders can exploit unilaterally.

Policy-based authentication frameworks — SPF, DKIM, and DMARC — represent essential, complementary layers of email security, and their continued deployment and enforcement must be encouraged. The 2025 adoption data, however, is sobering: 61.9% of the top ten million internet domains lack valid SPF records; 81.6% lack DMARC records; and of those with DMARC, only 3.9% enforce the most protective policy level. Documented bypass techniques — SMTP Smuggling (CVE-2023-51764), DKIM replay attacks, SPF forwarding failures, and dangling include domain abuse — demonstrate that even correctly deployed frameworks have exploitable failure modes. The fundamental limitation of all three is their dependence on decisions made by the sending domain operator, outside the defender's control.

The bottom-most `Received` header transcends this dependency. It is the only piece of metadata in a standard email message that was authored by the receiver rather than the sender, from a data source the sender cannot reach. It offers deterministic, binary, rule-encodable verification of network origin — a ground-truth complement to policy-based frameworks that cannot be nullified by sender misconfiguration or omission.

The security community's call to action is clear: email security vendors should surface this signal prominently in their tooling interfaces and APIs; email administrators should build extraction and comparison logic into inbound filter rules; security operations teams should elevate header chain analysis to the first step of every phishing investigation; and researchers should continue to refine methodologies for handling the legitimate edge cases — forwarding, shared hosting, IPv6 rDNS gaps — that introduce ambiguity. The mechanism has been present in every SMTP deployment since RFC 821. The only missing piece has been the systematic will to use it.

> *"The most powerful security controls are often those that exploit properties of the network infrastructure itself — properties the attacker cannot change, suppress, or negotiate away."*

---

10. References / Bibliography

26. **[1]** Klensin, J. (2008). *Simple Mail Transfer Protocol*. RFC 5321. Internet Engineering Task Force (IETF). https://www.rfc-editor.org/rfc/rfc5321
27. **[2]** Resnick, P. (Ed.). (2008). *Internet Message Format*. RFC 5322. Internet Engineering Task Force (IETF). https://www.rfc-editor.org/rfc/rfc5322
28. **[3]** Wong, M., & Schlitt, W. (2006). *Sender Policy Framework (SPF) for Authorizing Use of Domains in Email*. RFC 4408 (obsoleted). Kitterman, S. (2014). RFC 7208. IETF. https://www.rfc-editor.org/rfc/rfc7208
29. **[4]** Allman, E., Callas, J., Delany, M., Libbey, M., Fenton, J., & Thomas, M. (2007). *DomainKeys Identified Mail (DKIM) Signatures*. RFC 6376. IETF. https://www.rfc-editor.org/rfc/rfc6376
30. **[5]** Kucherawy, M., & Zwicky, E. (2015). *Domain-based Message Authentication, Reporting, and Conformance (DMARC)*. RFC 7489. IETF. https://www.rfc-editor.org/rfc/rfc7489
31. **[6]** Rekhter, Y., Moskowitz, B., Karrenberg, D., de Groot, G. J., & Lear, E. (1996). *Address Allocation for Private Internets*. RFC 1918. IETF. https://www.rfc-editor.org/rfc/rfc1918
32. **[7]** National Vulnerability Database (NVD). (2023). *CVE-2023-51764: Postfix SMTP Smuggling Authentication Bypass*. NIST. https://nvd.nist.gov/vuln/detail/CVE-2023-51764
33. **[8]** SEC Consult. (2023). *SMTP Smuggling — Spoofing E-Mails Worldwide*. https://sec-consult.com/blog/detail/smtp-smuggling-spoofing-e-mails-worldwide/

34. **[9]** Fortra. (2025). *Global DMARC Adoption Trends in Q2 2025*. Fortra Email Security Research. https://www.fortra.com/resources/reports/global-dmarc-adoption-trends

35. **[10]** EasyDMARC. (2025). *EasyDMARC 2025 DMARC Adoption Report*. https://easydmarc.com/blog/dmarc-adoption-report-2025

36. **[11]** Anti-Phishing Working Group (APWG). (2025). *Phishing Activity Trends Report — Q1 2025*. https://apwg.org/trendsreports/

37. **[12]** Interisle Consulting Group. (2025). *Phishing Landscape 2025: An Annual Study of the Scope and Distribution of Phishing*. https://interisle.net/PhishingLandscape2025.pdf

38. **[13]** Verizon. (2025). *2025 Data Breach Investigations Report (DBIR)*. https://www.verizon.com/business/resources/reports/dbir/

39. **[14]** Cisco Talos / SEC Consult. (2023). *SMTP Smuggling — Postfix Advisory and Mitigation*. http://www.postfix.org/smtp-smuggling.html

40. **[15]** Cymru, T. (2025). *BGP / ASN Reputation and IP Intelligence*. Team Cymru Community Services. https://www.team-cymru.com/ip-reputation

41. **[16]** AbuseIPDB. (2025). *IP Address Abuse Confidence Scoring*. https://www.abuseipdb.com

42. **[17]** VirusTotal. (2025). *IP Address and Domain Reputation Analysis*. https://www.virustotal.com

43. **[18]** Shodan. (2025). *Internet-Wide Port and Service Scanning Database*. https://www.shodan.io

44. **[19]** CISA. (2023). *Email Security Best Practices for Federal Agencies* (Binding Operational Directive 18-01 and successors). Cybersecurity and Infrastructure Security Agency. https://www.cisa.gov/email-security

45. **[20]** NIST. (2024). *NIST Special Publication 800-177 Rev. 1: Trustworthy Email*. National Institute of Standards and Technology. https://csrc.nist.gov/publications/detail/sp/800-177/rev-1/final

46. **[21]** Postel, J. (1982). *Simple Mail Transfer Protocol*. RFC 821 (obsoleted by RFC 5321). IETF. https://www.rfc-editor.org/rfc/rfc821

47. **[22]** Gont, F., & Bellovin, S. M. (2012). *Defending Against Sequence Number Attacks*. RFC 6528. IETF. https://www.rfc-editor.org/rfc/rfc6528
