

Lab: Investigating a Business Email Compromise (BEC) Scam

MSCS630: Cybercrime Investigation and Digital Evidence

Dr. Mouatez Karbab



September 10, 2026

1 Lab Overview and Scenario

You are a junior investigator assigned to the Cybercrime Unit of a corporate security team. The finance department has just forwarded a suspicious email to your team's incident response portal. An employee, Alice, received an email that appears to be from the company's CEO, requesting an urgent wire transfer to a new vendor to close a confidential deal. The amount is significant, and the request is unusual.

Initial checks with the CEO's office confirm that the CEO never sent this email. Your team suspects this is a classic Business Email Compromise (BEC) attempt, specifically a form of CEO fraud. Your task is to perform a preliminary technical analysis of the fraudulent email to trace its origin and gather evidence of its illegitimacy. You will analyse the email's full headers, which act as a digital "postal history," to uncover the path the message actually took across the internet and to identify the technical red flags that prove it is a forgery.

2 Learning Objectives

Upon successful completion of this lab, you will be able to:

- Understand the structure and purpose of email headers for digital investigations.
- Use online analysis tools to parse and visualise the path of an email.
- Trace an email's delivery path by reading the **Received:** headers in the correct (bottom-to-top) order, and explain why only the hops added by servers you trust can be relied upon.
- Identify and extract the originating IP address of a suspicious email.

- Use IP geolocation and WHOIS lookups to determine who operates a source IP address, and explain why that is not the same as locating the attacker.
- Identify key technical indicators of email spoofing, and distinguish which identity each of SPF and DKIM actually authenticates — neither of them, on its own, vouches for the **From:** address the recipient sees.
- Document technical findings in a structured, professional preliminary analysis report.

3 Prerequisites: Tools & Data Setup

This lab primarily uses web-based tools, so no complex software installation is required. You will need a modern web browser and an internet connection. A terminal is used only to hash the evidence file and, optionally, to run a WHOIS lookup.

3.1 Tool Installation: Web-Based Analysis Tools

No installation is needed for these tools. They are accessed directly through a web browser.

- **Google Admin Toolbox Messageheader:** Parses email headers and lays out the delivery path hop by hop, with the delay introduced at each one.
 - **Link:** <https://toolbox.googleapps.com/apps/messageheader/>
- **MXToolbox Email Header Analyzer:** A second parser, useful as a cross-check, which also resolves the sending domain's email-related DNS records.
 - **Link:** <https://mxtoolbox.com/EmailHeaders.aspx>
- **IP lookup:** A service that reports the approximate location of an IP address and, more importantly for an investigation, the network that owns it.
 - **Link:** <https://ipinfo.io/> (alternative: <https://www.geoiptool.com/>)
 - **CLI equivalent:** `whois 198.54.117.211`, which returns the registration record held by the regional internet registry, including the network's abuse contact.

3.2 Data Source: Sample BEC Email Header

For this investigation, you will use the full email header from the reported BEC email. This raw text contains all the metadata required to trace the message's journey.

1. Download/Create the Evidence File:

- Copy the entire block of text below, from **Delivered-To:** down to the final `</html>`.
- Paste it into a plain text editor (like Notepad on Windows, TextEdit on macOS in plain-text mode, or `nano` in a Linux terminal).
- Save the file as `bec_email_header.txt`. Save it with Unix (LF) line endings and a single trailing newline; the hash below will not match otherwise.

```
Delivered-To: alice.employee@example.com
Received: by 2002:a05:620a:10c8:0:0:0:0 with SMTP id z8csp2406822qkk;
        Fri, 13 Aug 2025 10:02:59 -0700 (PDT)
X-Google-Smtp-Source: ABCs/1234567890abcdefghijklmno=
X-Received: by 2002:a5d:43c4:0:b0:220:b922:a5a9 with SMTP id x4-20020
        a5d43c4000000b00220b922a5a9mr9988748wrp.89.1660399379017;
        Fri, 13 Aug 2025 10:02:59 -0700 (PDT)
```

```

ARC-Seal: i=1; a=rsa-sha256; t=1660399379; cv=none;
        d=google.com; s=arc-20160816;
        b=somehashvalue
ARC-Message-Signature: i=1; a=rsa-sha256; c=relaxed/relaxed; d=google.com;
        s=arc-20160816;
        h=to:subject:message-id:from:date:mime-version:dkim-signature;
        bh=anotherhashvalue;
        b=yetanotherhashvalue
ARC-Authentication-Results: i=1; mx.google.com;
        dkim=pass header.i=@ceo-offices.com header.s=google header.b=AbCdEfG
        ;
        spf=fail (google.com: domain of support@v-pshosting.net does not
        designate 198.54.117.211 as permitted sender) smtp.mailfrom=support@v-
        pshosting.net
Return-Path: <support@v-pshosting.net>
Received: from mail.v-pshosting.net (mail.v-pshosting.net.
        [198.54.117.211])
        by mx.google.com with ESMTP id a6-20020
        a170902c30600b001a1d50c1157si910115plg.437.2025.08.13.10.02.58
        for <alice.employee@example.com>;
        Fri, 13 Aug 2025 10:02:58 -0700 (PDT)
Received-SPF: fail (google.com: domain of support@v-pshosting.net does not
        designate 198.54.117.211 as permitted sender) client-ip=198.54.117.211;
Authentication-Results: mx.google.com;
        dkim=pass header.i=@ceo-offices.com header.s=google header.b=AbCdEfG
        ;
        spf=fail (google.com: domain of support@v-pshosting.net does not
        designate 198.54.117.211 as permitted sender) smtp.mailfrom=support@v-
        pshosting.net
DKIM-Signature: v=1; a=rsa-sha256; c=relaxed/relaxed; d=ceo-offices.com; s=
        google; h=mime-version:date:from:message-id:subject:to; bh=
        anotherhashvalue; b=somehashvalue
MIME-Version: 1.0
Date: Fri, 13 Aug 2025 17:02:55 +0000
From: "CEO John Doe" <john.doe@example.com>
Message-ID: <123456789.abscd@ceo-offices.com>
Subject: Urgent: Confidential Payment Request
To: alice.employee@example.com
Content-Type: text/html; charset=UTF-8

<html><body><p>Alice, I need you to process an urgent wire transfer for $15
,000 to vendor account 87654321. This is for the confidential Project X
. Please process immediately and confirm once sent. Discretion is key
.</p><p>Regards,<br>John Doe<br>CEO</p></body></html>

```

2. Verify Data Integrity:

- To confirm you have copied the text correctly, calculate its SHA256 hash. In a real investigation this is what proves the working copy is the evidence and that it has not been altered since it was collected.
- **BASH Command (Linux, macOS, WSL):**

```

1 sha256sum bec_email_header.txt
2

```

On macOS, `shasum -a 256 bec_email_header.txt` produces the same value.

- **Windows Command Prompt:** there is no `sha256sum` on Windows; use the built-in `certutil` instead.

```

1 certutil -hashfile bec_email_header.txt SHA256
2

```

- **Expected SHA256 Hash:**

```
ebabf124b0b1c4b747efd26f271f8b60101406be3b05b1bde8e22182504ae962
```

- *Observation/Expected Outcome:* If your hash does not match, do not proceed and do not "note the discrepancy" in the report — a copy that does not hash to the expected value is not the evidence, and nothing derived from it can be tied back to what was collected. Re-copy the block, check for a stray blank line at the end or for line endings converted to CRLF by your editor, and hash it again.

4 Lab Procedure: Step-by-Step Analysis

Follow these steps to trace the email and identify fraudulent indicators.

4.1 Step 1: Parse the Email Header

The first step is to use a specialised tool to parse the raw header into a readable form. This helps visualise the email's journey and flags authentication problems automatically. You will still verify each finding against the raw text afterwards: the tool tells you what the headers say, not whether they can be believed.

1. Copy the Header Data

Open your saved `bec_email_header.txt` file and copy its entire content to your clipboard (Ctrl+A, Ctrl+C or Cmd+A, Cmd+C).

2. Analyze with Google Admin Toolbox

- **GUI Instructions:**

- (a) Navigate to the **Google Admin Toolbox Messageheader** tool: <https://toolbox.googleapps.com/apps/messageheader/>.
- (b) Paste the copied header into the text box provided on the page.
- (c) Click the "**ANALYZE THE HEADER ABOVE**" button.

- *Observation/Expected Outcome:* The tool displays a summary table (message ID, subject, creation date, sender, recipient) followed by a hop-by-hop table with the delay introduced at each hop. Two things should stand out immediately. First, the chain is very short: there are only two **Received:** headers, so the message reached Alice's provider in a single external hop. Second, the summary reports an SPF failure. Note both; you will return to them in Steps 2 and 4.

4.2 Step 2: Trace the Email's Path

Each **Received:** header is added at the *top* of the message by the server that has just accepted it. The chain therefore reads bottom-to-top in chronological order: the bottom-most **Received:** line is the earliest hop recorded in the message, and the top-most is the last.

1. Examine the Received: Headers

- **GUI Instructions (in Messageheader tool):** Scroll below the summary table. The tool lists the hops in chronological order, so it has already reversed the raw header for you: **Hop 1 is the earliest** — the message arriving at `mx.google.com` from an external server — and the final row is the last internal handoff before delivery to Alice.
- *Manual Inspection:* In the raw text there are exactly two **Received:** lines. The lower one, **Received: from mail.v-pshosting.net ... by mx.google.com**, records the moment Google's mail server accepted the message from outside. The upper one, **Received: by 2002:a05:620a:10c8...**, is internal to Google's own infrastructure and tells you nothing about the sender.
- *Observation/Expected Outcome:* The hop you care about is the earliest one added by a server you trust — here, the line written by `mx.google.com`. That is the boundary of trust. Everything at or above it was recorded by infrastructure under the receiving provider's control and can be relied upon; anything below it would have been written by servers the attacker may control and could be entirely fabricated. In this message

there is nothing below it at all, which is itself informative: the sender injected the email directly into Google’s gateway rather than relaying it through a chain of hosts, so `mail.v-pshosting.net` is the first hop and the earliest sending infrastructure the message can attest to.

4.3 Step 3: Identify the Originating IP Address and Geolocation

The earliest trusted **Received:** header contains the IP address of the server that delivered the email to the mail gateway. This is your primary lead on the attacker’s infrastructure.

1. Extract the Originating IP

- **GUI Instructions:** In the Messageheader analysis, open **Hop 1** — the earliest hop. The tool labels the sending server’s name and its IP address.
- *Manual Inspection:* In the raw header, find this line:

```
Received: from mail.v-pshosting.net (mail.v-pshosting.net.  
[198.54.117.211])
```

The address in square brackets is the IP the receiving server actually saw the connection come from: `198.54.117.211`. Note that the hostname before the brackets is supplied by the connecting client and can be anything; the bracketed IP is observed by the receiving server and cannot be forged over a completed TCP connection. The **Received-SPF:** line records the same value independently as `client-ip=198.54.117.211`, which corroborates it.

2. Geolocate and Attribute the IP Address

- **GUI Instructions:**
 - (a) Copy the IP address `198.54.117.211`.
 - (b) Navigate to an IP lookup service such as <https://ipinfo.io/>.
 - (c) Paste the IP address into the search box and run the lookup.
- **CLI Instructions:** the registry record carries the information an investigator actually needs — the network’s owner and its abuse contact.

```
1 whois 198.54.117.211  
2
```

- *Observation/Expected Outcome:* The address belongs to a commercial hosting provider’s network in the United States, and reverse DNS resolves it to that provider’s own infrastructure rather than to `mail.v-pshosting.net`. Record the network owner and the abuse contact: they are the party you would have to serve with a preservation request. Be careful what you conclude from the city the tool reports. It is the location of a rented server, not of a person. Whoever sent this email could have been anywhere, reaching that host over SSH or a web panel, and the hosting account itself was very likely paid for with stolen or anonymous payment details.

4.4 Step 4: Identify Technical Red Flags of Forgery

Now correlate the evidence to build the case that this email is fraudulent. Look for contradictions between the identities the message asserts and the identities that were actually verified.

1. Analyze the From: Address vs. The Envelope Sender

- **From: Header:** The email claims to be from "CEO John Doe" <john.doe@example.com>. This is the only sender identity the recipient's mail client displays, and it is written by the sender — nothing verifies it by default.
- **Return-Path / smtp.mailfrom:** The envelope sender, recorded in **Return-Path:** <support@v-pshosting.net> and repeated as **smtp.mailfrom** in the authentication results, is an address at **v-pshosting.net**.
- *Observation/Expected Outcome:* The displayed sender and the envelope sender are in different domains. SMTP permits this legitimately — mailing lists and bulk senders rely on it — so a mismatch is not proof of forgery on its own. Combined with a request for an urgent, confidential, unverifiable wire transfer from a CEO whose office denies sending it, it is a strong red flag, and it is the reason the authentication results below matter.
- **Corroborating detail:** the Message-ID is <123456789.abscd@ceo-offices.com>. A message genuinely composed on **example.com**'s mail system would carry a Message-ID in the **example.com** domain, generated by that system. This one names a third domain — the same one the DKIM signature belongs to — which places the message's origin outside the company entirely.

2. Check Email Authentication Results (SPF & DKIM)

- **SPF (Sender Policy Framework):** SPF checks whether the connecting IP is authorised to send mail for the domain in the *envelope* sender — not for the **From:** address the user sees. Find the result in the **Received-SPF:** header and again in **Authentication-Results**.
- **GUI Instructions (in Messageheader tool):** the tool displays the SPF result prominently, with a red "Fail" label.
- **Raw Header:** the header states **spf=fail** and gives the reason:

```
domain of support@v-pshosting.net does not designate 198.54.117.211 as
permitted sender
```

The operator of **v-pshosting.net** has published an SPF record, and 198.54.117.211 is not in it. So the sending host was not authorised even for the throwaway domain it put on the envelope — the attacker either did not control that domain's DNS or did not bother to configure it.

- **DKIM (DomainKeys Identified Mail):** the result is **dkim=pass** header.i=@ceo-offices.com, and the **DKIM-Signature** header shows **d=ceo-offices.com**. Read that carefully, because a "pass" is easy to misread as a verdict of legitimacy. All it establishes is that whoever signed the message controls DNS for **ceo-offices.com** and that the signed headers were not modified in transit. It says nothing whatsoever about **example.com**. **ceo-offices.com** is a lookalike domain the attacker registered; a valid signature from it is trivially obtained, since anyone can sign mail for a domain they own.
- *Observation/Expected Outcome:* The decisive finding is one of *alignment*, and it is what DMARC evaluates. The **From:** domain is **example.com**. SPF authenticated **v-pshosting.net** — and failed. DKIM authenticated **ceo-offices.com** — and passed. Neither authenticated **example.com**. Not one identity check in this message vouches for the domain shown to Alice, so the claim that the CEO sent it has no technical support at all. That, not the SPF failure by itself, is what makes the message provably a forgery.

5 Lab Report

Complete the following report template based on your analysis. This document simulates a preliminary report you would provide to your incident response manager. Write it in your own words: every value you need is one you produced in Steps 1 to 4, and a finding stated without the evidence that supports it is not a finding.

Cybercrime Investigation - Preliminary Analysis Report

Case ID: *LAB-02-BEC-[Your-Name]*

Investigator: *[Your Name]*

Date of Analysis: *[Today's Date]*

1. Evidence Details

- **Filename:** *bec_email_header.txt*
- **SHA256 Hash:** *[Paste the SHA256 hash value you calculated during data setup]*
- **Message Timestamps:** *[The time in the **Date:** header and the time recorded by the receiving gateway in the earliest trusted **Received:** header. Give both in UTC and say whether they agree.]*

2. Executive Summary

[Three or four sentences, written for a manager who will not read the rest of the report: what was attempted, against whom, by what mechanism, and why identifying the person responsible is not straightforward. No header-level detail here — that belongs in Section 3.]

3. Key Findings

- **Attack Type Identified:** *[Name the attack type, then give the evidence that establishes it. Two well-chosen observations are enough.]*
- **Target:** *[This attack targeted a person, not a host, so there is no victim IP address to report. Identify the recipient and the finance process the request was trying to subvert.]*
- **Attack Source:** *[The originating IP from Step 3, the network that owns it, and its reverse DNS. Note that unlike a spoofed-source flood this address is genuine — the TCP connection completed, so it is the real address of the sending host — and say what that does and does not tell you about the attacker.]*
- **Key Indicators:**
 - **Technical:** *[SPF result: _____ on domain _____. DKIM result: _____ on signing domain _____. The **From:** domain: _____. Originating IP: _____. Number of **Received:** hops: _____.]*

- **Alignment:** *[In one sentence: which identity did each authentication check actually verify, and why does neither result support the **From:** address the recipient saw?]*
- **Qualitative:** *[The social-engineering markers in the message body and subject — urgency, confidentiality, authority, a request to bypass normal verification — and the header mismatches that corroborate them.]*

4. Attribution Challenges

[Given what you established about the source, what can and cannot be concluded about who sent this email? Address the fact that the originating IP identifies rented hosting rather than a person, that the sending host may itself be compromised, and that the domains involved were registered for this purpose. What would the next investigative step be — whose logs, whose records, and through what legal mechanism — and how quickly would it have to be taken?]