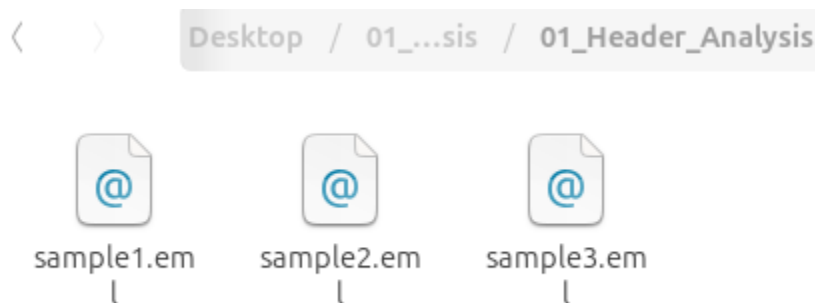


Header Analysis

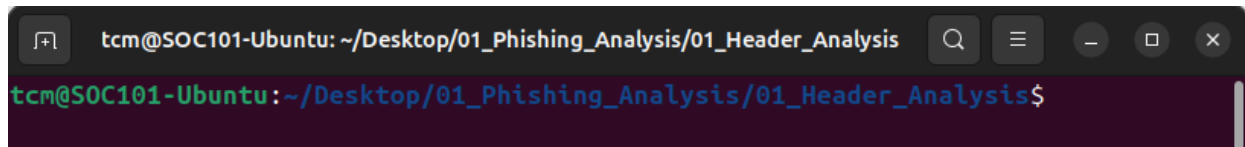
Email headers contain valuable forensic information that can help identify malicious emails and trace their origins. This section covers the tools and techniques needed to analyze email headers effectively, including command-line methods, text editors, and online analyzers. Understanding email authentication mechanisms (SPF, DKIM, DMARC) is crucial for determining whether an email is legitimate or potentially malicious.

Opening email files

The course provided 3 sample eml files:

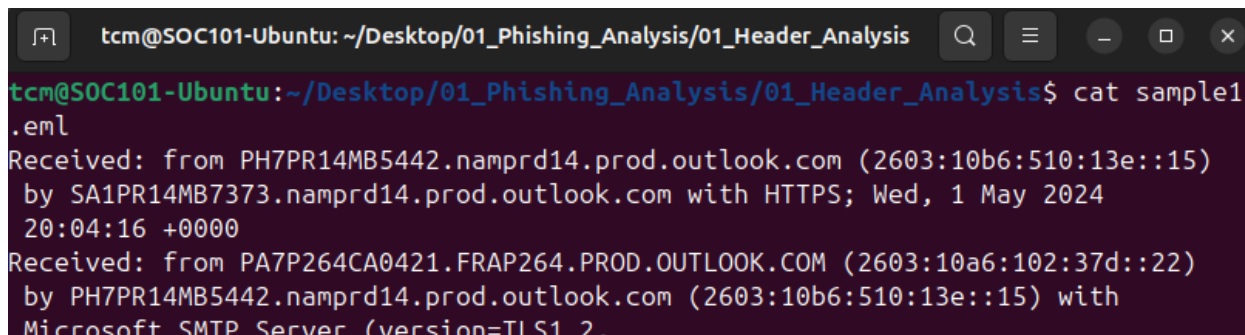


Right click the folder and open in terminal to easily navigate to this folder in the terminal:



As mails and eml files are fundamentally just text files, you can cat them to read the text, or grep them to find specific field such as From:

Cat example:

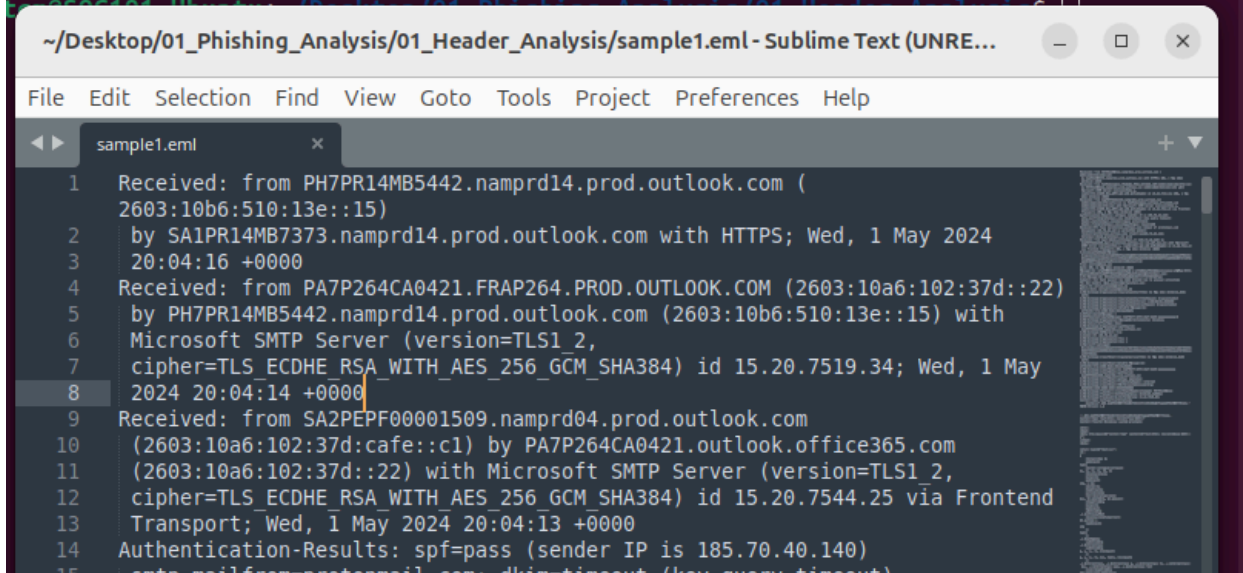


Grep example:

```
tcm@SOC101-Ubuntu: ~/Desktop/01_Phishing_Analysis/01_Header_Analysis
tcm@SOC101-Ubuntu:~/Desktop/01_Phishing_Analysis/01_Header_Analysis$ grep From sample1.eml
From: alerts@chase.com
X-MS-Exchange-CrossTenant-FromEntityHeader: Internet
tcm@SOC101-Ubuntu:~/Desktop/01_Phishing_Analysis/01_Header_Analysis$
```

Working with a large text file in the terminal is not ideal, so we can open it in a text editor like Sublime instead:

```
tcm@SOC101-Ubuntu: ~/Desktop/01_Phishing_Analysis/01_Header_Analysis
tcm@SOC101-Ubuntu:~/Desktop/01_Phishing_Analysis/01_Header_Analysis$ subl sample1.eml
```



The screenshot shows the Sublime Text editor window titled "sample1.eml - Sublime Text (UNRE...)". The menu bar includes File, Edit, Selection, Find, View, Goto, Tools, Project, Preferences, and Help. The file "sample1.eml" is open, displaying an email header. The text is as follows:

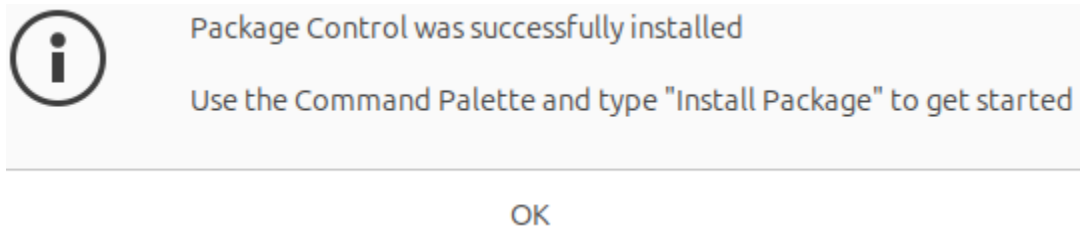
```
1 Received: from PH7PR14MB5442.namprd14.prod.outlook.com (
2 2603:10b6:510:13e::15)
3 by SA1PR14MB7373.namprd14.prod.outlook.com with HTTPS; Wed, 1 May 2024
4 20:04:16 +0000
5 Received: from PA7P264CA0421.FRAPH264.PROD.OUTLOOK.COM (2603:10a6:102:37d::22)
6 by PH7PR14MB5442.namprd14.prod.outlook.com (2603:10b6:510:13e::15) with
7 Microsoft SMTP Server (version=TLS1 2,
8 cipher=TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384) id 15.20.7519.34; Wed, 1 May
9 2024 20:04:14 +0000
10 Received: from SA2PEPF00001509.namprd04.prod.outlook.com
11 (2603:10a6:102:37d:cafe::c1) by PA7P264CA0421.outlook.office365.com
12 (2603:10a6:102:37d::22) with Microsoft SMTP Server (version=TLS1 2,
13 cipher=TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384) id 15.20.7544.25 via Frontend
14 Transport; Wed, 1 May 2024 20:04:13 +0000
15 Authentication-Results: spf=pass (sender IP is 185.70.40.140)
16 smtp.mailfrom=protonmail.com; dkim=timeout (key query timeout)
```

Highlighting headers in sublime

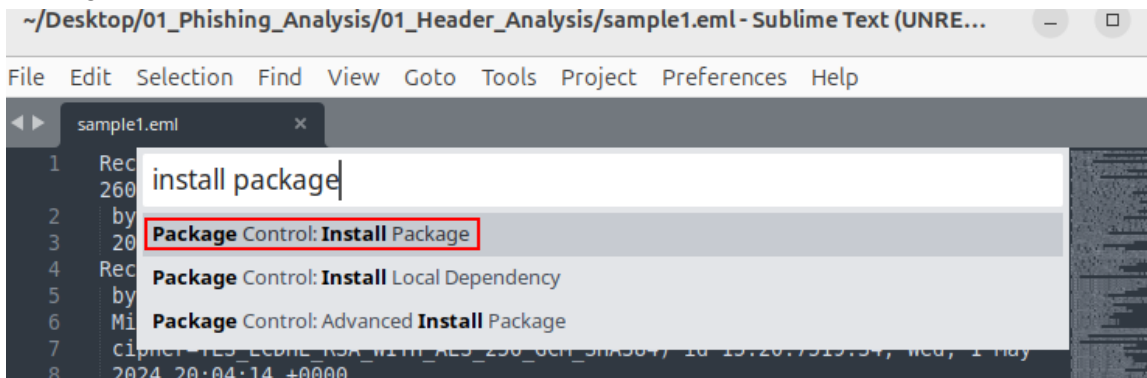
The course recommends to install a syntax highlighter by 13 Cubed

Steps:

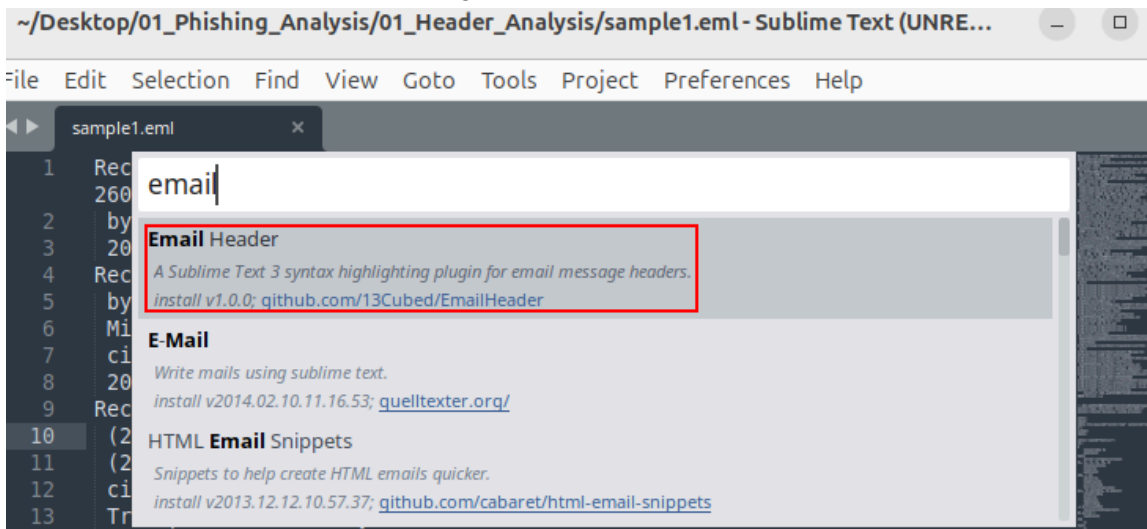
1. In Sublime, go to Tools > Install Package Control...
You should get a box confirming Package Control was installed:



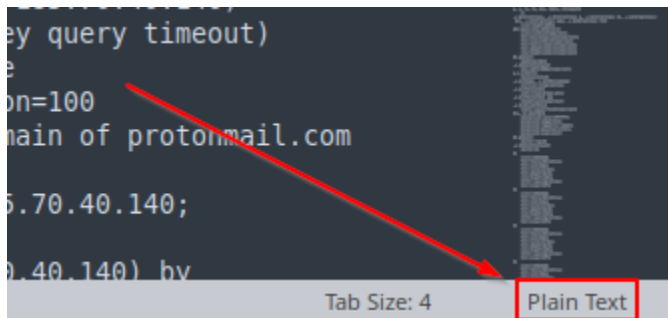
2. Next use Ctrl+Shift+p to open the command palet, look for Package Control: Install Package, and click on it.



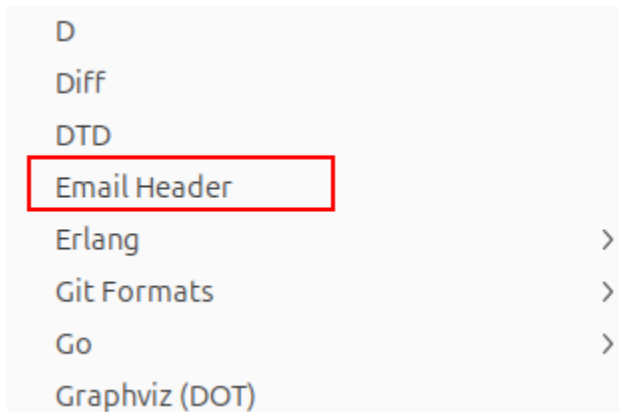
3. Search for the Email Header package and click on it



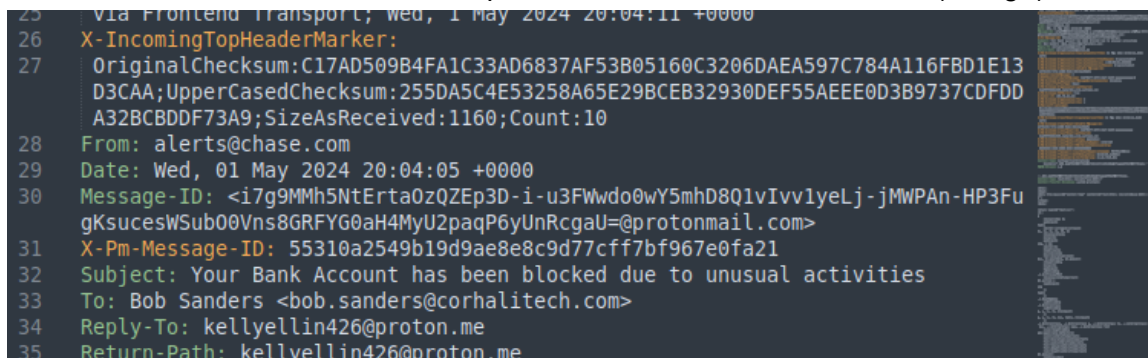
- Now you have it installed, click on the menu in the lower right:



And you'll have an option in the lower right menu called Email Header.



- The headers names are now highlighted in various colors. This makes it easier to read and to find certain headers, and to separate them from custom headers (orange).



Alternatively, you can use <http://mha.azurewebsites.net> to analyze headers. This is a client side analysis, so information is not shared with any servers.

Simply copy the text from the eml file into the text box and click Analyze headers to get a table with header information.

Message Header Analyzer x

+

←

→

↻

🔒

mha.azurewebsites.net

☆

🔔

📄

☰

Message Header Analyzer

— Insert the message header you would like to analyze

Received: from PH7PR14MB5442.namprd14.prod.outlook.com (2603:10b6:510:13e::15) by SA1PR14MB7373.namprd14.prod.outlook.com with HTTPS; Wed, 1 May 2024 20:04:16 +0000

Received: from PA7P264CA0421.FRAP264.PROD.OUTLOOK.COM (2603:10a6:102:37d::22) by PH7PR14MB5442.namprd14.prod.outlook.com (2603:10b6:510:13e::15) with Microsoft SMTP Server (version=TLS1_2, cipher=TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384) id 15.20.7519.34; Wed, 1 May 2024 20:04:14 +0000

Received: from SA2PEPF00001509.namprd04.prod.outlook.com (2603:10a6:102:37d:cafe::c1) by PA7P264CA0421.outlook.office365.com (2603:10a6:102:37d::22) with Microsoft SMTP Server (version=TLS1_2, cipher=TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384) id 15.20.7544.25 via Frontend Transport; Wed, 1 May 2024 20:04:13 +0000

Authentication-Results: spf=pass (sender IP is 185.70.40.140) smtp.mailfrom=protonmail.com; dkim=timeout (key query timeout)

Analyze headers

Clear

Copy

Submit feedback on github

— Summary

Subject Your Bank Account has been blocked due to unusual activities

Message Id <I7g9MMh5NiErtaOzQE3D-Iu3FWwdo0wY5mhD8Q1vIvw1yeLj-JMWPAn-HP3FugKsuceWSSubOOVns8GRFYG0aH4MyU2paqP6yUnRcgaU=@protonmail.com>

Creation time Wed, 01 May 2024 20:04:05 +0000 (Delivered after 5 seconds)

From alerts@chase.com

Reply to kellyellin426@proton.me

To Bob Sanders <bob.sanders@corhalitech.com>

— Received headers

| Hop | Submitting host | Receiving host | Time | Delay | Type |
|-----|--|---|---------------------|-----------|--|
| 1 | mail-40140.protonmail.ch (185.70.40.140) | SA2PEPF00001509.mail.protection.outlook.com (10.167.242.41) | 5/1/2024 9:04:11 PM | | Microsoft SMTP Server (version=TLS1_3, cipher=TLS_AES_256_GCM_SHA384) |
| 2 | SA2PEPF00001509.namprd04.prod.outlook.com (2603:10a6:102:37d:cafe::c1) | PA7P264CA0421.outlook.office365.com (2603:10a6:102:37d::22) | 5/1/2024 9:04:13 PM | 2 seconds | Microsoft SMTP Server (version=TLS1_2, cipher=TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384) |
| 3 | PA7P264CA0421.FRAP264.PROD.OUTLOOK.COM (2603:10a6:102:37d::22) | PH7PR14MB5442.namprd14.prod.outlook.com (2603:10b6:510:13e::15) | 5/1/2024 9:04:14 PM | 1 second | Microsoft SMTP Server (version=TLS1_2, cipher=TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384) |
| 4 | PH7PR14MB5442.namprd14.prod.outlook.com (2603:10b6:510:13e::15) | SA1PR14MB7373.namprd14.prod.outlook.com (172.16.17.1) | 5/1/2024 9:04:16 PM | 2 seconds | HTTPS |

There is a similar tool at <https://mxtoolbox.com/EmailHeaders.aspx>, but this processes the information server side, as there are more checks that it does. For example, it checks signatures and DKIM authentication.

Email headers

The course recommends this page from iana (Internet Assigned Numbers Authority) to look up standardized message headers:

<https://www.iana.org/assignments/message-headers/message-headers.xhtml>

The website shows standard, obsolete, experimental and other headers. The course will focus on the most common and useful headers.

Date: This is the sending time and date. This is important for timing and records keeping.

From: Specifies the supposed sender of the email. This can be set to any arbitrary value and can therefore be spoofed rather easily. Can still be used to search for other phishes using that sender email, and group them together.

Subject: This is the subject line of the email, often seen in email processors as a kind of title. Can also be used to search for similar phishes and group them together, or to block emails with certain subject lines.

Message-ID: This should be a unique ID for that email, generated by the sender's email client or the first mail server in the sending domain. A kind of fingerprint. Once assigned, it shouldn't change as the message travels. Duplicate message IDs are a likely indicator that malicious activity is going on.

There are two parts: before the @ is the unique fingerprint, after the @ is the host/domain where it was created. If the domain differs from the sender domain, this could also hint at malicious activity.

To: Recipient email address. Useful for determining scope. As in, who is being targeted? How many? Note: even if 1 recipient, BCC might have been used to send to multiple targets

Reply-To: Specifies where replies to the email should go to. Usually the same as From. While there are legitimate reasons to have a different Reply-To address, it can also indicate email spoofing. After all, the attacker would not have access to the legitimate From: email address if it was spoofed.

Return-Path: Also known as the envelope sender address or bounce address. It specifies where email bounce messages or delivery failure emails should go to. Again, usually the same as From, and if this is different, it could mean malicious activity. In some cases, with mass phishing campaigns, where the attacker does not care about reconnaissance, the attacker might have this field be different from Reply-To, to make sure they are not spammed with bounce emails.

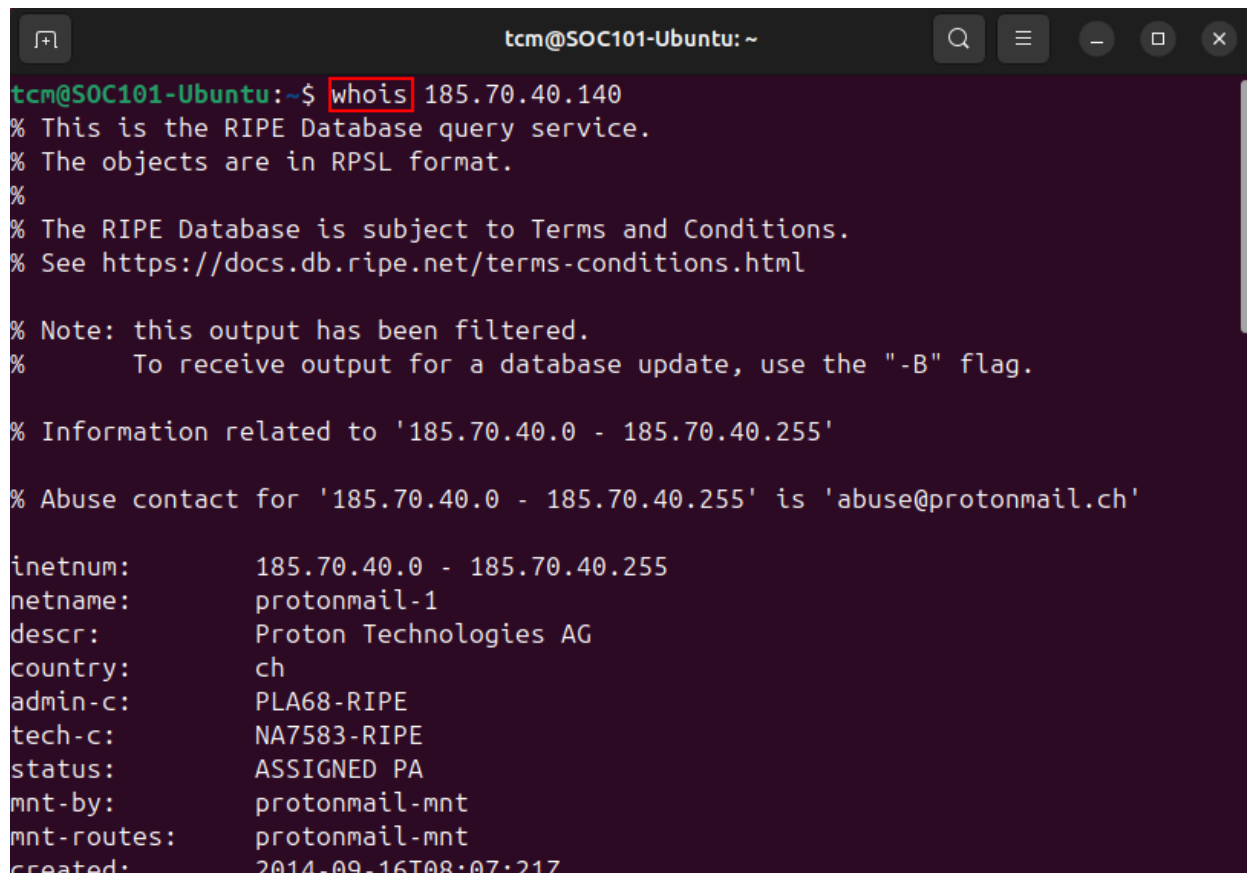
X-type headers: Experimental or custom headers. They do not fit in the standards adopted by the IANA. They are sometimes used by email providers or security solutions to add additional functionality to email clients and filters. Think spam checks or tracking or authentication results.

X-Sender-IP or X-Originating-IP: This is one of the more common X-headers, but not always present. Indicates the IP of the device or server where the message originated from. Can be used for geolocation, , who owns it, what reputation the IP has, reverse DNS lookup. Note: sometimes this shows the IP of an intermediate server instead of the end-user.

Received: Most useful. There are multiple, as the email goes through multiple MTAs as it is being routed. Also called the Received-chain. Each MTA adds its own header with its own information, such as IP. A bit like a passport. Get a new stamp every time you travel to another country. They are in reverse chronological order (top is most recent). Start from the bottom to tell the route that the email took. The top one should be the recipient's mail server. Note that some of the earlier ones could have been spoofed. Only trust the ones from MTA that are trusted.

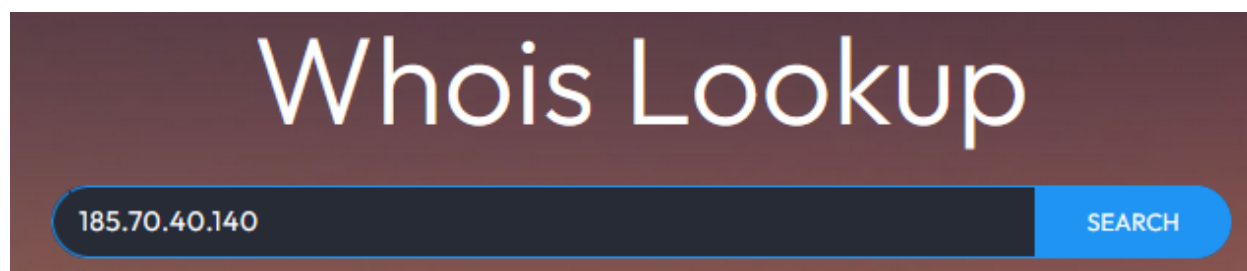
Authentication-Results: This lists whether the SPF, DKIM and DMARC checks passed.

Use whois in a terminal to determine who owns an IP:



```
tcm@SOC101-Ubuntu: ~  
tcm@SOC101-Ubuntu:~$ whois 185.70.40.140  
% This is the RIPE Database query service.  
% The objects are in RPSL format.  
%  
% The RIPE Database is subject to Terms and Conditions.  
% See https://docs.db.ripe.net/terms-conditions.html  
  
% Note: this output has been filtered.  
%       To receive output for a database update, use the "-B" flag.  
  
% Information related to '185.70.40.0 - 185.70.40.255'  
  
% Abuse contact for '185.70.40.0 - 185.70.40.255' is 'abuse@protonmail.ch'  
  
inetnum:        185.70.40.0 - 185.70.40.255  
netname:        protonmail-1  
descr:          Proton Technologies AG  
country:        ch  
admin-c:        PLA68-RIPE  
tech-c:         NA7583-RIPE  
status:         ASSIGNED PA  
mnt-by:         protonmail-mnt  
mnt-routes:     protonmail-mnt  
created:        2014-09-16T08:07:21Z
```

Alternatively, use <http://whois.domaintools.com>
Simply type in the IP address:





Whois Lookup

185.70.40.140

And then you get the information:

IP Information for 185.70.40.140

— Quick Stats

| | |
|--------------|--|
| IP Location |  Switzerland Plan-les-ouates Proton Technologies Ag |
| ASN |  AS62371 PROTON Proton AG, CH (registered Nov 17, 2015) |
| Resolve Host | 185-70-40-140.protonmail.ch |
| Whois Server | whois.ripe.net |
| IP Address | 185.70.40.140 |

```
% Abuse contact for '185.70.40.0 - 185.70.40.255' is ' abuse@protonmail.ch '

inetnum:        185.70.40.0 - 185.70.40.255
netname:        protonmail-1
descr:          Proton Technologies AG
country:        ch
admin-c:        PLA68-RIPE
tech-c:         NA7583-RIPE
status:         ASSIGNED PA
mnt-by:         protonmail-mnt
mnt-routes:     protonmail-mnt
created:        2014-09-16T08:07:21Z
last-modified:  2024-07-31T12:58:35Z
source:         RIPE

role:           Proton NOC
address:        Route de la Galaise 32
address:        1228 Plan-les-Ouates
```

Email Authentication methods

There are three technologies to authenticate emails: SPF, DKIM, DMARC. The header called Authentication-Results shows whether checks have passed or not.

SPF (Sender Policy Framework)

Allows domain owners to specify which mail servers are authorized to send emails on behalf of their domain. They do so by publishing SPF records in the Domain Name System (DNS). The SPF records contain a list of IP addresses that are authorized to send emails for their domain.

You can look this up with nslookup or dig

```
Nslookup -type=txt {domain} | grep spf
```



```
tcm@S0C101-Ubuntu:~$ nslookup -type=txt shodan.io | grep spf
shodan.io      text = "v=spf1 ip4:216.117.2.180 ip4:69.72.37.146 include:_spf.g
oogle.com -all"
```

Dig TXT {domain} | grep -i spf

```
tcm@S0C101-Ubuntu:~$ dig TXT shodan.io | grep -i spf
shodan.io.      62      IN      TXT      "v=spf1 ip4:216.117.2.180 ip4:69
.72.37.146 include:_spf.google.com -all"
```

SPF records contain the version, usually version 1 (v=spf1), a list of ip addresses (ip4:216.117.2.180) and what other domains are allowed to send emails on behalf of the domain, in this case include everything in google's spf records as well (include:_spf.[google.com](https://www.google.com)). The -all at the end means everyone else is not allowed to send emails on behalf of shodan. ~all means "softfail" - it suggests that emails from unlisted sources should probably be rejected, but leaves the final decision to the receiving server.

So when an email is received, the receiving mail server looks up the SPF record of the domain using a DNS query. If the IP address or domain of the sending mail server matches something in the SPF records, then it passes. The domain that is checked is the one in the Return-Path header.

Note: Passing an SPF check does not verify whether the sender is legitimate or not. All it does is check whether the sender matches or is authorized by the SPF records. If a scammer uses Gmail or creates their own SPF records for their own domain, then it will pass these checks.

DKIM (DomainKeys Identified Mail)

Used to authenticate the origin of email messages using PKI (public key infrastructure). Can verify whether an email was sent by the domain it claims to be from. And also confirms whether the message has been tampered with during transit.

When an email is sent from a domain with DKIM enabled, the sending mail server adds a signature to the email header using a private key (shows up as DKIM-Signature: {signature}). The recipient can then look up the public key of that domain using the sending domain's DNS server to verify that the message was actually from that domain and was not altered.

Only checks whether the domain is legitimate or not, and whether the email has been tampered with since sending. It cannot determine whether the sender or domain is malicious or not. For example, an attacker could create a typosquatted domain similar to a legitimate organization and set up DKIM for that lookalike domain, or the attacker could use a hacked email address that is legitimate.

DKIM-Signature: v={version number}; a={hashing algorithm}; c=relaxed/relaxed; d={domain}; s={selector}; bh={bodyhash}; b={DKIM signature}

Public key can be found with the selector and domain using nslookup.

Nslookup -type=txt {selector}._domainkey.{domain}

```
tcm@S0C101-Ubuntu:~$ nslookup -type=txt s1._domainkey.namecheap.com
;; Truncated, retrying in TCP mode.
Server:          127.0.0.53
Address:         127.0.0.53#53

Non-authoritative answer:
s1._domainkey.namecheap.com      canonical name = s1.domainkey.u1828068.wl069.sendgrid.net.
s1.domainkey.u1828068.wl069.sendgrid.net      text = "k=rsa; t=s; p=MIIBIjANBg
kqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA4EJ2WbK3G12fhP8hLHBTABlvdbKePJXwux+sJGXRnnoVdG
Aaw9q9D96qeW3uWqAbBSyPB06w4zTeK1qi7Ar+rBC91zKEiuoi6Rbd8xkDBG1Emo8RMhZj0Her5x10To
bynvYy6J4F/ge40gA17nNdfc7n2Xg+00KHVY4dVZfdgNR29eGraxD8X0E2pMBdNgtqKvt6S" "4irlnE
uhvko+Ls3XqBicTnM30Q04ffYIJWLUqHEwVjBUHKXV+/sTif8UecWw2m9uLYLPbeNBajMcRtmKYC+tKT
39laA2mtPuQub9LHtgzkmAXqE9D7uvgc8gEoUgdvQyefKCLRR/rKomB9CeQIDAQAB"
```

The public key is the long string after p=

You can manually look up the DKIM public key using <http://mxtoolbox.com/dkim.aspx>.

 **DKIM Record Lookup**

Domain Name

Selector

DKIM Lookup

Or analyze the headers using <https://mxtoolbox.com/EmailHeaders.aspx> and it will do the DKIM check for you.

| | Test | Result |
|---|-----------------------------------|--|
| ✖ | DKIM Signature Body Hash Verified | Body Hash Did Not Verify |
| ✔ | DKIM Record Published | DKIM Record found |
| ✔ | DKIM Syntax Check | The record is valid |
| ✔ | DKIM Public Key Check | Public key is present |
| ✔ | DKIM Signature Syntax Check | The signature is valid |
| ✔ | DKIM Signature Identifier Match | Signature domain match |
| ✔ | DKIM Signature Alignment | Signature domain in alignment. |
| ✔ | DKIM Signature Duplicate Tags | Signature tags are unique |
| ✔ | DKIM Signature Expiration | The signature is not expired |
| ✔ | DMARC Record Published | DMARC Record found |
| ✔ | DMARC Policy Not Enabled | DMARC Quarantine/Reject policy enabled |

In this case the body hash check did not go through, which means something was altered in the body of the email between sending and receiving. (The course redacted some information, so this is expected in this case.)

DMARC (Domain-based Message Authentication, Reporting & Conformance)

Works alongside SPF and/or DKIM. Has additional reporting mechanisms and adds a layer of control and visibility for domain owners. Domain owners can specify policies on what to do with an email when an SPF or DKIM check fails or the email is not aligned. None: don't do anything. Quarantine: spam it. Reject: removes it. Note that these are just policies on the domain owner side, and recipients email servers don't have to honor them if they do not want to. Same with reporting, the recipient does not have to report to a domain whether they received emails from that domain.

Reporting: Aggregate reports contain summary statistics, not individual emails. These give domain owners visibility into their email ecosystem and potential abuse. Forensic reports contain samples of individual emails that failed DMARC. Again, it is up to the recipient's email server to decide how much they send and what conforms to their privacy policy.

Alignment: DMARC checks whether the From header matches either the domain in the Return-Path, or the domain in the DKIM signature.

Summary

Email header analysis involves examining metadata to trace email origins and verify authenticity. Key tools include terminal commands (cat, grep), text editors with syntax highlighting, and online analyzers like MHA and MXToolbox. The most important headers for security analysis are Received (shows email routing), Authentication-Results (shows SPF/DKIM/DMARC status), and various sender-related fields. While authentication mechanisms help verify domain legitimacy, they cannot determine if the sender or domain itself is malicious.

Red Flags Checklist

- ☐ Message-ID domain differs from sender domain
- ☐ Reply-To address differs from From address without legitimate reason
- ☐ Return-Path differs from From address (especially in mass campaigns)
- ☐ Multiple Received headers from suspicious or unrecognized servers

- ☐ Failed SPF, DKIM, or DMARC authentication checks
- ☐ Suspicious IP addresses in X-Sender-IP or X-Originating-IP headers
- ☐ Inconsistent routing in the Received chain
- ☐ Custom X-headers that seem unusual or suspicious
- ☐ Timestamps that don't align with expected sending patterns