Understanding MIME types and how they impact email communication and security

If you have ever sent or received an email with an attachment, you have likely encountered MIME types, though you may not have realized it. MIME types play a crucial role in modern email communication. For IT professionals and advanced users, understanding MIME types is essential not only for troubleshooting but also for ensuring security. Let's break down what MIME types are, their history, and why they matter.

What are MIME types?

MIME stands for **Multipurpose Internet Mail Extensions**. MIME types are identifiers used to describe the format of a file or the nature of the data contained in an email or a web page. They are essential for enabling email systems and web servers to recognize and handle different types of content appropriately.

For example, a file with the MIME type `image/png` is a PNG image, while `application/pdf` refers to a PDF document.

A MIME type consists of two parts, separated by a slash:

1. ***Type*:** The general category, like `text`, `image`, `audio`, or `application`.
2. ***Subtype*:** The specific format, like `html`, `jpeg`, or `json`.

Why was MIME invented?

In the early days of email (1980s), the format of email messages was limited to plain text using the ASCII character set. This meant users couldn't send files, images, or non-

English characters. As the internet grew, so did the need for richer and more versatile communication.

MIME was introduced in 1991 by Nathaniel Borenstein and his colleagues at the Carnegie Mellon University as an extension to the original email protocol (SMTP) to address these limitations. MIME made it possible to:

- Attach files like images, documents, and audio to emails.
- Include non-ASCII text (e.g., characters from other languages).
- Support multipart messages (e.g., combining plain text and HTML versions of an email).

MIME is an Internet standard, specified in a series of RFC: RFC 2045, RFC 2046, RFC 2047, RFC 4288, RFC 4289 and RFC 2049. The integration with SMTP email is specified in RFC 1521 and RFC 1522.

MIME headers in email

Most email clients add one or several of the following MIME headers in the sent emails to describe the content type and format.

- ***MIME-Version*:** specifies the MIME-Version; currently set at 1.0 and will likely stay as is forever.
- ***Content-Type*:** specifies the MIME type of the message part. For example:
  - `text/plain; charset=UTF-8`: plain text with UTF-8 encoding.
  - `multipart/mixed`: a message with multiple parts (e.g., text and attachments).
- ***Content-Disposition*:** indicates if a part should be displayed inline or treated as an attachment.
- ***Content-Transfer-Encoding*:** defines how the content is encoded for safe transmission (e.g., base64 for binary data).

Main MIME types used in email

Here are some common MIME types you'll encounter in email communication:

- **Text-Based Content**
  - `text/plain`: Plain text emails.
  - `text/html`: HTML emails with rich formatting.
- **Images**
  - `image/jpeg`: JPEG images.
  - `image/png`: PNG images.
- **Documents**

  - `application/pdf`: PDF files.
  - `application/msword`: Microsoft Word documents.
- **Audio and Video**
  - `audio/mpeg`: MP3 audio files.
  - `video/mp4`: MP4 video files.
- **Compressed Files**
  - `application/zip`: ZIP archives.
  - `application/gzip`: GZIP files.
- **Executables**
  - `application/x-executable`: Binary executables.
  - `application/x-msdownload`: Windows executable files.

You can find an extensive list of MIME types on the freeformatter.com site.

How do email clients handle MIME types?

When an email client receives a MIME-formatted message, it processes the headers to determine how to display the content. For example:

- A `text/html` part will render as a rich email with images and formatting.
- An `image/jpeg` part with a `Content-Disposition : inline` will appear embedded in the email body.
- An `application/pdf` part with a `Content-Disposition: attachment` will be shown as an attachment or downloadable file for web-based mail clients.

Modern email clients are designed to handle complex MIME structures, such as nested multiparts, ensuring that users see the email content as intended.

Example 1: alternative format

The following example shows an email whose content can be displayed as a simple text or as an HTML page, depending on the client capabilities or settings.

```
MIME-Version: 1.0
Date: Wed, 27 Nov 2024 17:08:05 +0100
Message-ID: <CAEjb-qPTn6j_nQTMuFazpwuDYrEZxFRhGNa-
3JYACmShH6HKCw@mail.gmail.com>
Subject: Test email
From: pl_trash <pltrash2@gmail.com>
To: pl_trash <pltrash2@gmail.com>
Content-Type: multipart/alternative;
boundary="000000000000c95da50627e72ba1"
```

```
--000000000000c95da50627e72ba1
Content-Type: text/plain; charset="UTF-8"

Test email.

--000000000000c95da50627e72ba1
Content-Type: text/html; charset="UTF-8"

<div dir="ltr"><div>Test email.</div><div><br></div></div>

--000000000000c95da50627e72ba1--
```

- The ***Content-Type: multipart/alternative*** header tells the email client that the mail contains multiple parts containing the same information in different forms.
- The ***boundary=*** qualifier lists the string used to separate the different parts of the multipart body.
- The first part is the email content in plain text (***Content-Type: text/plain; charset="UTF-8"***)
- The second part is the email content in HTML (***Content-Type: text/html; charset="UTF-8"***)

Depending on its capabilities, the email client will choose to display the plain text or the HTML version.

Example 2: mail with attachment

The following example shows an email with a separate file attachment.

```
MIME-Version: 1.0
Date: Tue, 10 Dec 2024 16:23:24 +0100
Message-ID: <CAEjb-qPc2h1hf5_2vGp8FMW6B-K5nUx1m_2W3EVsEORmg4D-
wg@mail.gmail.com>
Subject: Test mail with attachment.
From: pl_trash <pltrash2@gmail.com>
To: pl_trash <pltrash2@gmail.com>
Content-Type: multipart/mixed;
boundary="000000000000e51e3e0628ec0fa7"

--000000000000e51e3e0628ec0fa7
Content-Type: multipart/alternative;
boundary="000000000000e51e3a0628ec0fa5"

--000000000000e51e3a0628ec0fa5
Content-Type: text/plain; charset="UTF-8"

Test mail with attachment.
```

```
--000000000000e51e3a0628ec0fa5
Content-Type: text/html; charset="UTF-8"

<div dir="ltr"><div>Test mail with
attachment.</div><div><br></div></div>

--000000000000e51e3a0628ec0fa5--
--000000000000e51e3e0628ec0fa7
Content-Type: text/plain; charset="US-ASCII";
name="Test_attachment.txt"
Content-Disposition: attachment; filename="Test_attachment.txt"
Content-Transfer-Encoding: base64
X-Attachment-Id: f_m4im2j340
Content-ID: <f_m4im2j340>

VGVzdCBhdHRhY2htZW50IGZpbGUuCg==
--000000000000e51e3e0628ec0fa7--
```

- The first header, ***Content-Type: multipart/mixed***, is the way to signal an email with attachment.
- The second header, ***Content-Type: multipart/alternative***, as in the previous example, signals the same content in alternative form. Note that the separator string is different from the on separating the email body and the attachment.
- The 2 following headers, ***Content-Type: text/plain*** and ***Content-Type: text/html***, introduce the 2 versions of the email body
- The 5th header, ***Content-Type: text/plain; charset="US-ASCII"; name="Test_attachment.txt"***, specifies that the content of the attachment is a text file.
- ***Content-Disposition: attachment; filename="Test_attachment.txt"*** instructs the email client to handle the following content as an attachment, with a filename of *Test_attachment.txt*.
- ***Content-Transfer-Encoding: base64*** indicates that the content of the attachment has been encoded to the **base64** format. This algorithm allows any binary (non printable) content to be transformed in regular ASCII so it is safely transferred across channels that only support 7-bit ASCII characters (like early SMTP). This allows to include any type of file like pictures or executable files in an email attachment. The complete explanation of this encoding is outside of the scope of this article but will be addressed in detail in a future issue.
- Finally, the content of the attached file is ***VGVzdCBhdHRhY2htZW50IGZpbGUuCg==***, which is the base64 encoding of the string "***Test attachment file.***".

Why is it important to know about MIME types for security?

Understanding MIME types is critical for ensuring email security, as attackers often exploit them to deliver malicious content. Here's how:

1. Phishing and malware delivery

Attackers may disguise malicious files by using deceptive MIME headers. For example, a file named *document.pdf* may have a MIME type of `application/x-msdownload`, indicating it's actually an executable file.

2. Content Spoofing

Some malicious emails include mixed content, such as HTML emails with embedded scripts or hidden links. Recognizing unexpected MIME types can help identify such threats.

3. File Exploits

Certain file types (e.g., `.docx` or `.xlsx`) can contain macros or embedded code. By inspecting MIME types, users can flag and block potentially dangerous attachments.

4. Bypassing Email Filtering

Email security systems often rely on MIME types to filter spam and harmful content. Misconfigured MIME handling can allow malicious files to bypass these filters.

Conclusion

For IT professionals and advanced users, understanding MIME types is more than just a technical detail, it's a necessity for managing and securing email communication. By knowing how MIME works, you can:

- Diagnose issues with email formatting or attachments.
- Recognize potentially harmful content before it reaches end-users.
- Educate others on safe email practices.

In today's landscape of increasing email-based threats, this knowledge empowers you to stay one step ahead of attackers. Next time you inspect an email header, take a moment to appreciate the role of MIME in making modern communication possible and secure.