

SMTP and ncat

<https://github.com/heig-vd-dai-course>

[Web](#) • [PDF](#)

L. Delafontaine and H. Louis, with the help of GitHub Copilot.

Based on the original course by O. Liehti and J. Ehrensberger.

This work is licensed under the [CC BY-SA 4.0](#) license.

Objectives

- Learn electronic messaging protocols:
 - SMTP
 - POP3
 - IMAP
- Focus on the SMTP protocol
- Learn how to use ncat and Java to send an email to an SMTP server



Electronic messaging protocols: SMTP, POP3 and IMAP

More details for this section in the [course material](#). You can find other resources and alternatives as well.

Electronic messaging protocols: SMTP, POP3 and IMAP

- Email clients are called **Mail User Agents (MUA)**
- Email servers are called **Mail Transfer Agents (MTA)**
- They use different protocols to communicate



SMTP

- SMTP: Simple Mail Transfer Protocol
- Uses TCP port 25 (unencrypted) or 465 (encrypted)
- Used to send emails

The screenshot shows the IETF Datatracker page for RFC 5321, titled "Simple Mail Transfer Protocol". The page is from the Network Working Group, authored by J. Klensin in October 2008. It is a Request for Comments (RFC) 5321, which obsoletes RFC 2821 and is updated by RFC 1123. The category is "Standards Track".

Simple Mail Transfer Protocol

Status of This Memo

This document specifies an Internet standards track protocol for the Internet community, and requests discussion and suggestions for improvements. Please refer to the current edition of the "Internet Official Protocol Standards" (STD 1) for the standardization state and status of this protocol. Distribution of this memo is unlimited.

Abstract

This document is a specification of the basic protocol for Internet electronic mail transport. It consolidates, updates, and clarifies several previous documents, making all or parts of most of them obsolete. It covers the SMTP extension mechanisms and best practices for the contemporary Internet, but does not provide details about particular extensions. Although SMTP was designed as a mail transport and delivery protocol, this specification also contains information that is important to its use as a "mail submission" protocol for "split-UA" (User Agent) mail reading systems and mobile environments.

Document type

RFC Draft Standard

October 2008

[View errata](#)

[Report errata](#)

Updated by [RFC 7504](#)

Obsoletes [RFC 2821](#)

Updates [RFC 1123](#)

Was [draft-klensin-rfc2821bis](#) (individual in app area)

Select version

00	01	02	03	04
05	06	07	08	09
10	11	RFC 5321		

Compare versions

...nsin-rfc2821bis-10

RFC 5321

[Side-by-side](#) [Inline](#)

Author

[Dr. John C. Klensin](#)

POP3

- POP3: Post Office Protocol
- Uses TCP port 110 (unencrypted) or 995 (encrypted)
- Used to retrieve emails from a server

The screenshot displays the IETF Datatracker page for RFC 1939, 'Post Office Protocol - Version 3'. The page is organized into several sections:

- Header:** Includes the Network Working Group information, the request for comments (1939), the standard number (53), the obsolescence status (1725), and the category (Standards Track). It also lists the authors, J. Myers and M. Rose, and their affiliations, Carnegie Mellon and Dover Beach Consulting, Inc., with the date May 1996.
- Status:** Indicates the document is an Internet Standard.
- Document type:** Specifies the document is an RFC, an Internet Standard, dated May 1996.
- Table of Contents:** A detailed list of sections and their corresponding page numbers, including Introduction, A Short Digression, Basic Operation, The AUTHORIZATION State, The TRANSACTION State, The UPDATE State, Optional POP3 Commands, Scaling and Operational Considerations, POP3 Command Summary, Example POP3 Session, Message Format, References, Security Considerations, Acknowledgements, Authors' Addresses, and Appendix A.
- Navigation:** Includes links for 'View errata', 'Report errata', and 'Updated by' (RFC 1957, RFC 2449, RFC 6186, RFC 8314). It also lists obsolescence (RFC 1725) and alternative names (STD 53, draft-myers-pop3-pop3).
- Select version:** A dropdown menu showing the current version (RFC 1939) and other versions (00, 01, 02, 03, 04, 05, 06).
- Compare versions:** A section for comparing different versions of the document, with a dropdown for the current version (RFC 1939) and buttons for 'Side-by-side' and 'Inline' comparison.
- Authors:** Lists the authors, Dr. Marshall T. Rose and John G. Myers, with email addresses.

IMAP

- IMAP: Internet Message Access Protocol
- Uses TCP port 143 (unencrypted) or 993 (encrypted)
- Used to retrieve emails from a server
- Much more powerful than POP3 (synchronization, ...)

The screenshot shows the official Internet Message Access Protocol (IMAP) RFC 3501 page. The page is titled "INTERNET MESSAGE ACCESS PROTOCOL - VERSION 4rev1" and is categorized as a "Proposed Standard". It was authored by M. Crispin at the University of Washington in March 2003. The page includes a "Status of this Memo" section, a "Copyright Notice" (Copyright (C) The Internet Society (2003). All Rights Reserved.), and an "Abstract" section. The abstract describes the protocol's purpose: to allow a client to access and manipulate electronic mail messages on a server. It also mentions that the protocol supports a single server and does not specify a means of posting mail. The right sidebar contains a "Document type" section (RFC 3501, Proposed Standard, March 2003), a "View errata" button, a "Report errata" button, and a "Select version" table. The table shows a grid of version numbers from 00 to 19, with the current version (RFC 3501) highlighted in blue. Below the table is a "Compare versions" link.

Network Working Group
Request for Comments: 3501
Obsoletes: [2060](#)
Category: Standards Track

M. Crispin
University of Washington
March 2003

INTERNET MESSAGE ACCESS PROTOCOL - VERSION 4rev1

Status of this Memo

This document specifies an Internet standards track protocol for the Internet community, and requests discussion and suggestions for improvements. Please refer to the current edition of the "Internet Official Protocol Standards" (STD 1) for the standardization state and status of this protocol. Distribution of this memo is unlimited.

Copyright Notice

Copyright (C) The Internet Society (2003). All Rights Reserved.

Abstract

The Internet Message Access Protocol, Version 4rev1 (IMAP4rev1) allows a client to access and manipulate electronic mail messages on a server. IMAP4rev1 permits manipulation of mailboxes (remote message folders) in a way that is functionally equivalent to local folders. IMAP4rev1 also provides the capability for an offline client to resynchronize with the server.

IMAP4rev1 includes operations for creating, deleting, and renaming mailboxes, checking for new messages, permanently removing messages, setting and clearing flags, [RFC 2822](#) and [RFC 2045](#) parsing, searching, and selective fetching of message attributes, texts, and portions thereof. Messages in IMAP4rev1 are accessed by the use of numbers. These numbers are either message sequence numbers or unique identifiers.

IMAP4rev1 supports a single server. A mechanism for accessing configuration information to support multiple IMAP4rev1 servers is discussed in [RFC 2244](#).

IMAP4rev1 does not specify a means of posting mail; this function is handled by a mail transfer protocol such as [RFC 2821](#).

Datatracker [Info](#) [Contents](#) [Prefs](#)

RFC 3501
Proposed Standard

Document type
RFC **Proposed Standard**
March 2003
[View errata](#)
[Report errata](#)
Obsoleted by [RFC 9051](#)
Updated by [RFC 4466](#),
[RFC 4469](#), [RFC 4551](#),
[RFC 5032](#), [RFC 5182](#),
[RFC 5738](#), [RFC 6186](#),
[RFC 6858](#), [RFC 7817](#),
[RFC 8314](#), [RFC 8437](#),
[RFC 8474](#), [RFC 8996](#)
Obsoletes [RFC 2060](#)
Was [draft-crispin-imapv](#)
(individual in app area)

Select version

00	01	02	03	04
05	06	07	08	09
10	11	12	13	14
15	16	17	18	19
20	RFC 3501			

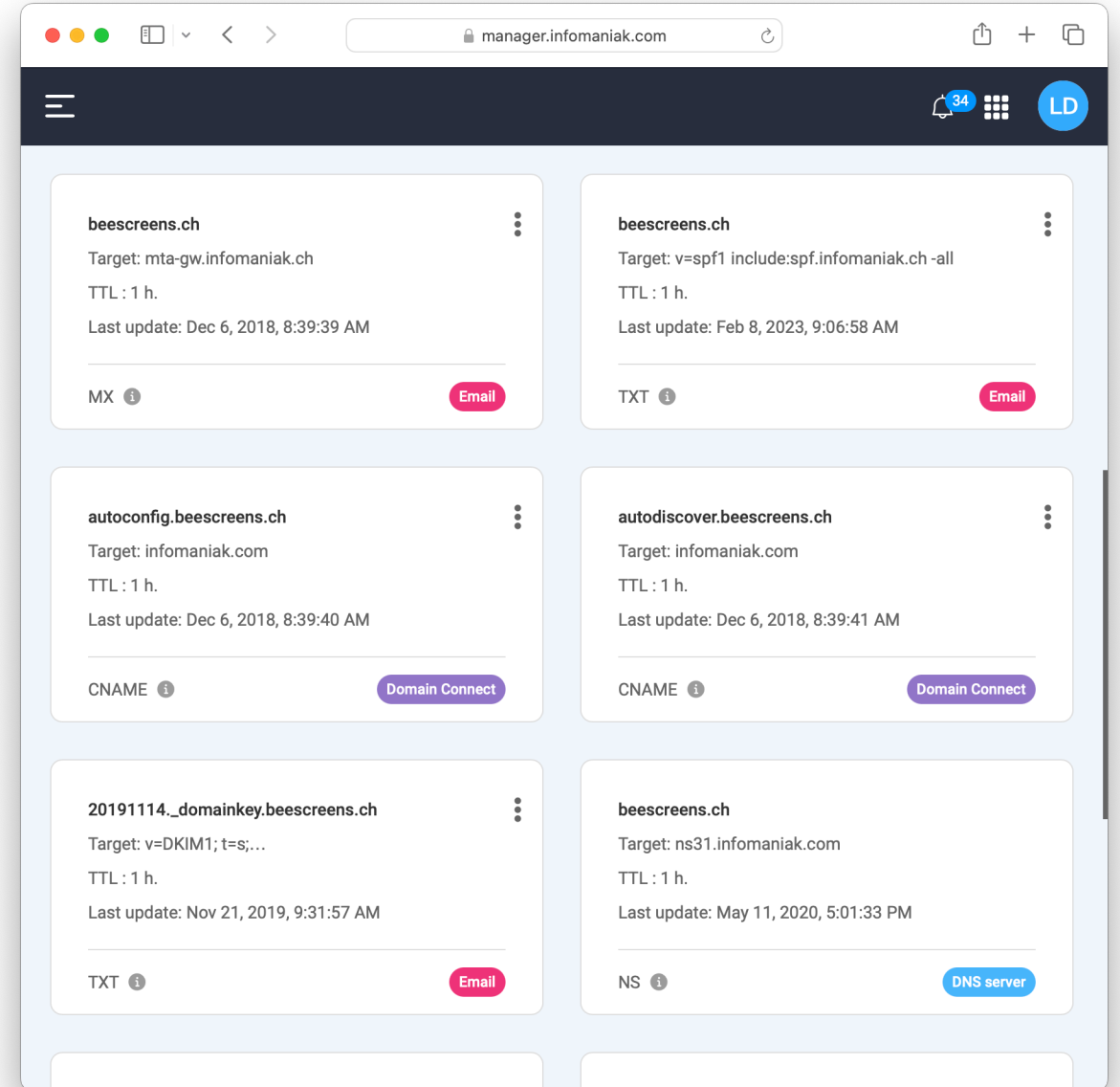
Compare versions

DNS records related to email

More details for this section in the [course material](#). You can find other resources and alternatives as well.

DNS records related to email

- **MX** : Mail eXchange - Specifies the mail server responsible for a domain name
- **TXT** : Store any text-based information. Used for **SPF** records, for email authentication

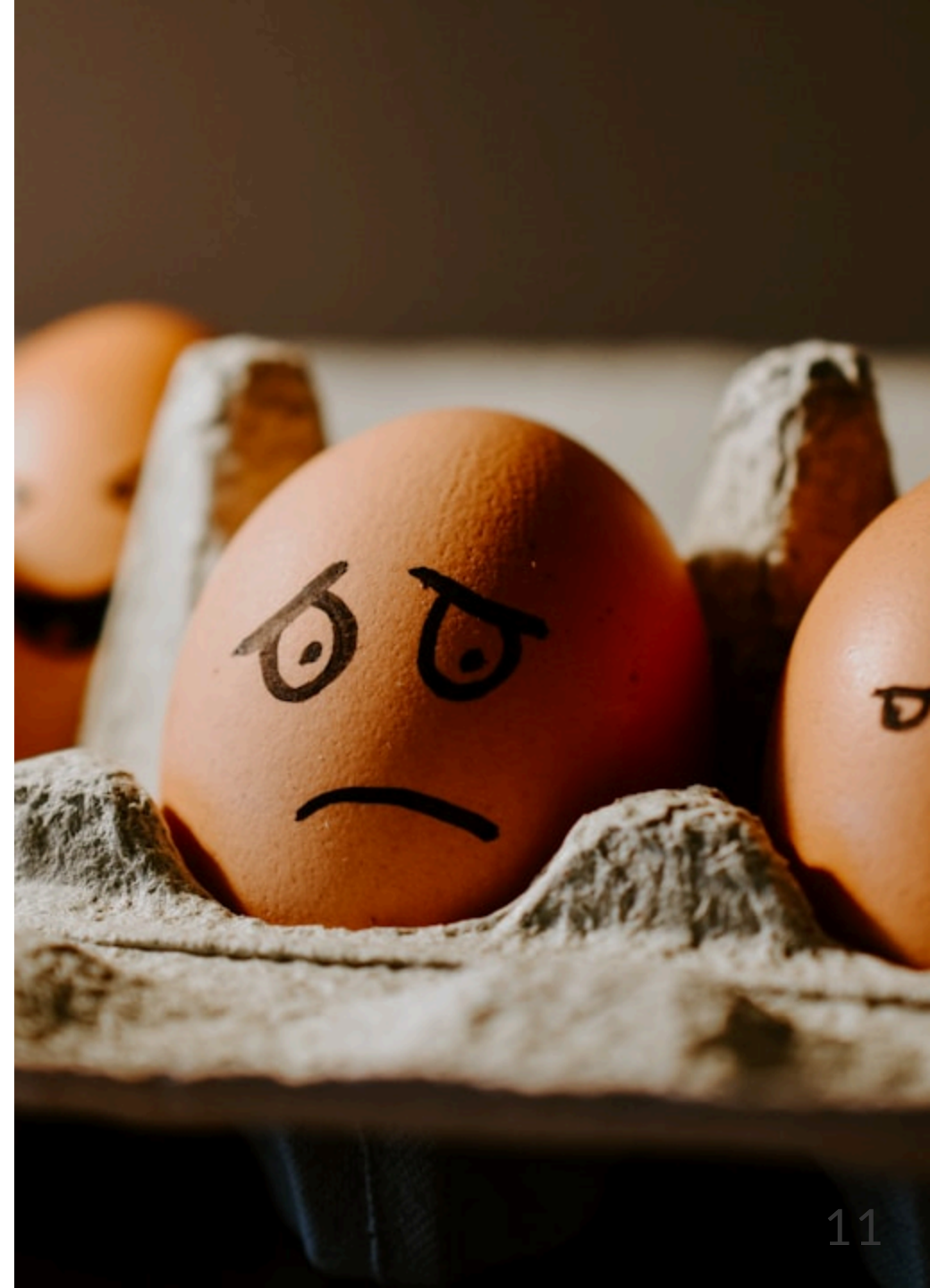


Security concerns and spam

More details for this section in the [course material](#). You can find other resources and alternatives as well.

Security concerns and spam

- SMTP is old and insecure
- Easy to spoof and forge emails
- Hard to maintain
- ➡ Your email server can be used for spam and can be blocked
- ➡ We will use a mock server to simulate an email server



A focus on the SMTP protocol

More details for this section in the [course material](#). You can find other resources and alternatives as well.

A focus on the SMTP protocol

- SMTP is a text-based protocol
- Messages are sent by the client to the server
- The server responds with a status code
- The client can send the next command

The screenshot displays the IETF Datatracker interface for RFC 5321, titled "SMTP". The page is divided into a main content area and a right-hand sidebar. The main content area shows the protocol specification, including the "Attdl-Protocol" section and the "4.5. Additional Implementation Issues" section, which further details the "4.5.1. Minimum Implementation" requirements. The sidebar on the right provides metadata about the document, including its type (RFC), draft status, date (October 2008), and links to errata, updates, and the author (Dr. John C. Klensin). A version selection table is also present, showing the progression from RFC 2821 to RFC 5321. The bottom of the page indicates the document is part of the "Standards Track" and is page 61 of the document.

PROTOCOL = "ESMTP" / "SMTP" / ATTDL-PROTOCOL

Attdl-Protocol = Atom

- ; Additional standard names for protocols are
- ; registered with the Internet Assigned Numbers
- ; Authority (IANA) in the "mail parameters"
- ; registry [9]. SMTP servers SHOULD NOT
- ; use unregistered names.

4.5. Additional Implementation Issues

4.5.1. Minimum Implementation

In order to make SMTP workable, the following minimum implementation MUST be provided by all receivers. The following commands MUST be supported to conform to this specification:

- EHLO
- HELO
- MAIL
- RCPT
- DATA
- RSET
- NOOP
- QUIT
- VERFY

Any system that includes an SMTP server supporting mail relaying or delivery MUST support the reserved mailbox "postmaster" as a case-insensitive local name. This postmaster address is not strictly necessary if the server always returns 554 on connection opening (as described in [Section 3.1](#)). The requirement to accept mail for postmaster implies that RCPT commands that specify a mailbox for postmaster at any of the domains for which the SMTP server provides mail service, as well as the special case of "RCPT TO:<Postmaster>" (with no domain specification), MUST be supported.

SMTP systems are expected to make every reasonable effort to accept mail directed to Postmaster from any other system on the Internet. In extreme cases -- such as to contain a denial of service attack or other breach of security -- an SMTP server may block mail directed to Postmaster. However, such arrangements SHOULD be narrowly tailored so as to avoid blocking messages that are not part of such attacks.

Klensin Standards Track [Page 61]

Datatracker

RFC 5321

Draft Standard

Info Contents Prefs

Document type

RFC **Draft Standard**

October 2008

[View errata](#)

[Report errata](#)

Updated by [RFC 7504](#)

Obsoletes [RFC 2821](#)

Updates [RFC 1123](#)

Was [draft-klensin-rfc2821bis](#) (individual in app area)

Select version

00	01	02	03	04
05	06	07	08	09
10	11	RFC 5321		

Compare versions

...nsin-rfc2821bis-10

RFC 5321

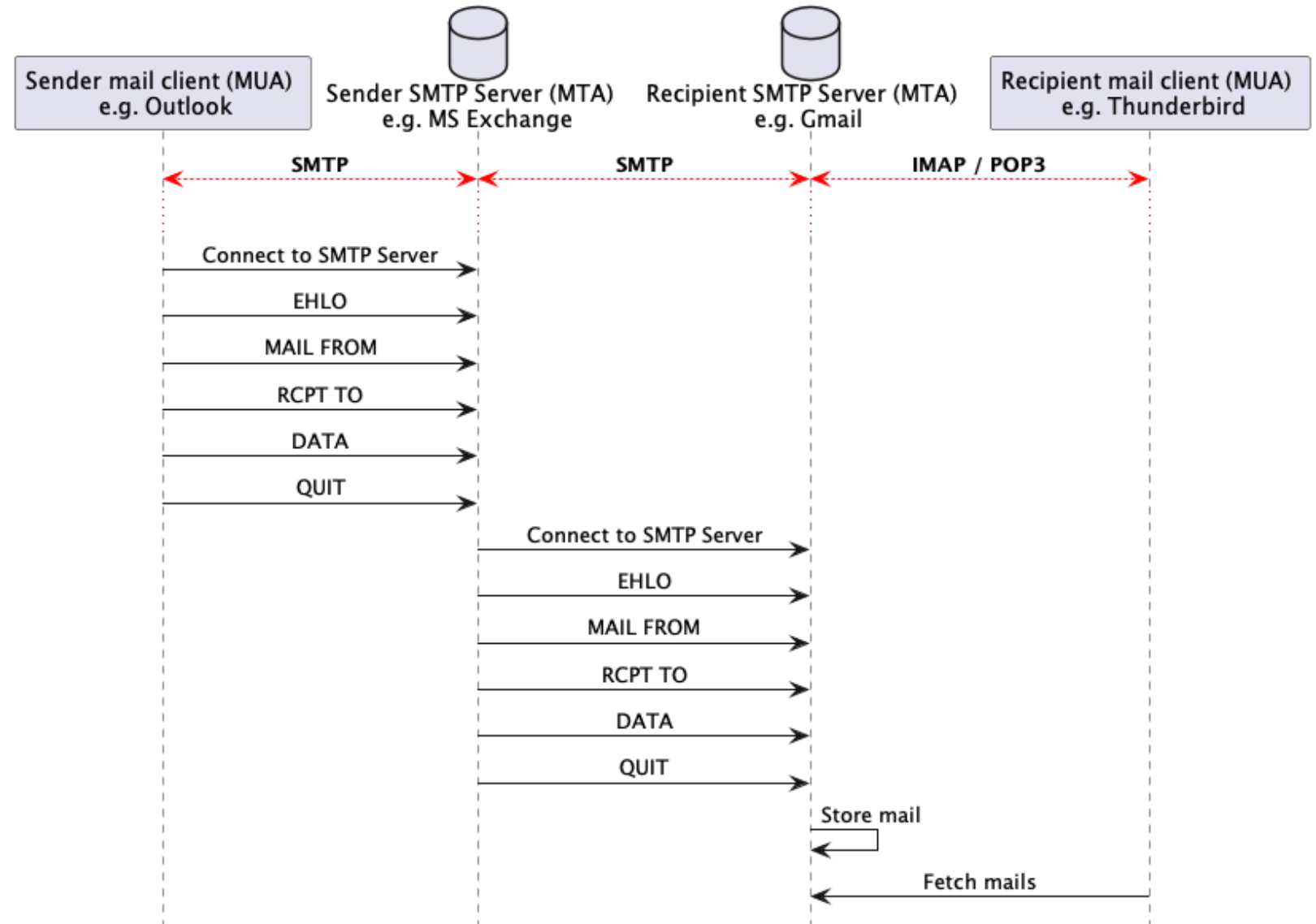
[Side-by-side](#) [Inline](#)

Author

[Dr. John C. Klensin](#) ✉

[Email author](#)

- HELO / EHLO
- MAIL FROM
- RCPT TO
- DATA
 - Subject:
 - From:
 - To:
 - End by .
- QUIT

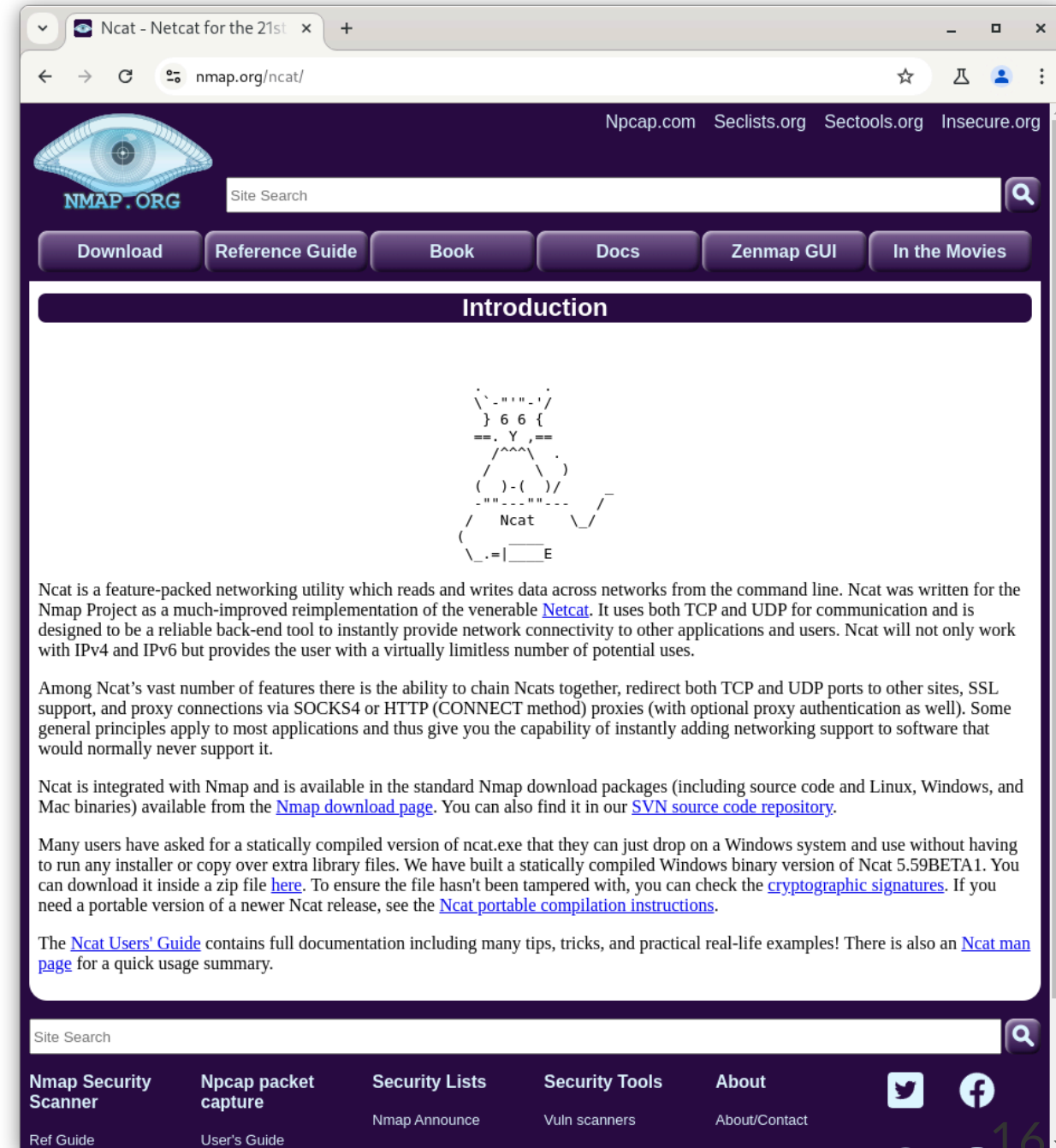


ncat

More details for this section in the [course material](#). You can find other resources and alternatives as well.

ncat

- ncat is network utility for reading from and writing to network connections
- It is used to connect to a remote server (SMTP, HTTP, ...)
- We will use it to interact with a SMTP server



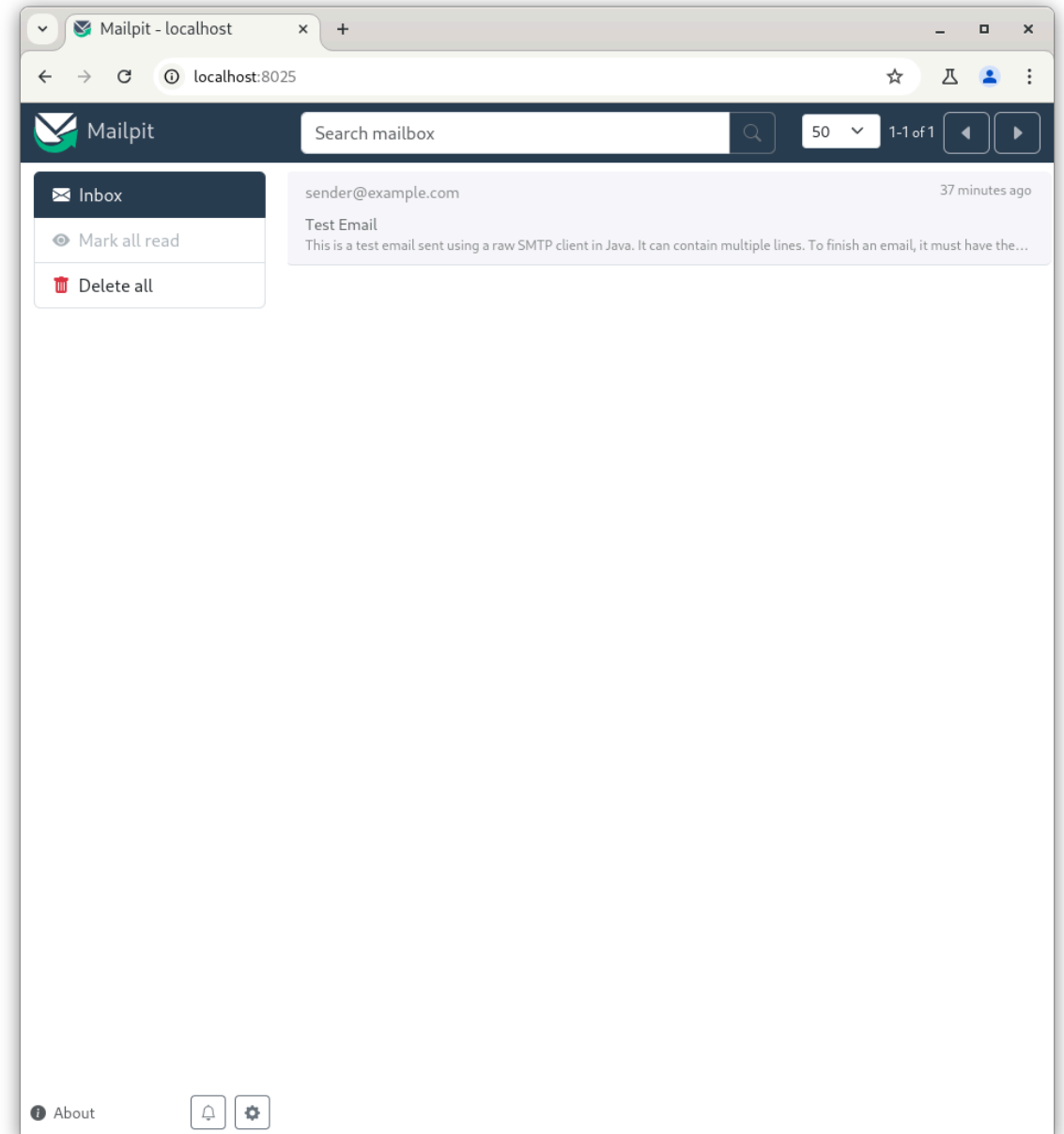
Questions

Do you have any questions?

Practical content

What will you do?

- Install and configure ncat
- Start a SMTP server with Docker Compose
- Send an email with ncat to the SMTP server
- Send an email with Java to the SMTP server



Find the practical content

You can find the practical content for this chapter on [GitHub](#).



Finished? Was it easy? Was it hard?

Can you let us know what was easy and what was difficult for you during this chapter?

This will help us to improve the course and adapt the content to your needs. If we notice some difficulties, we will come back to you to help you.

 [GitHub Discussions](#)

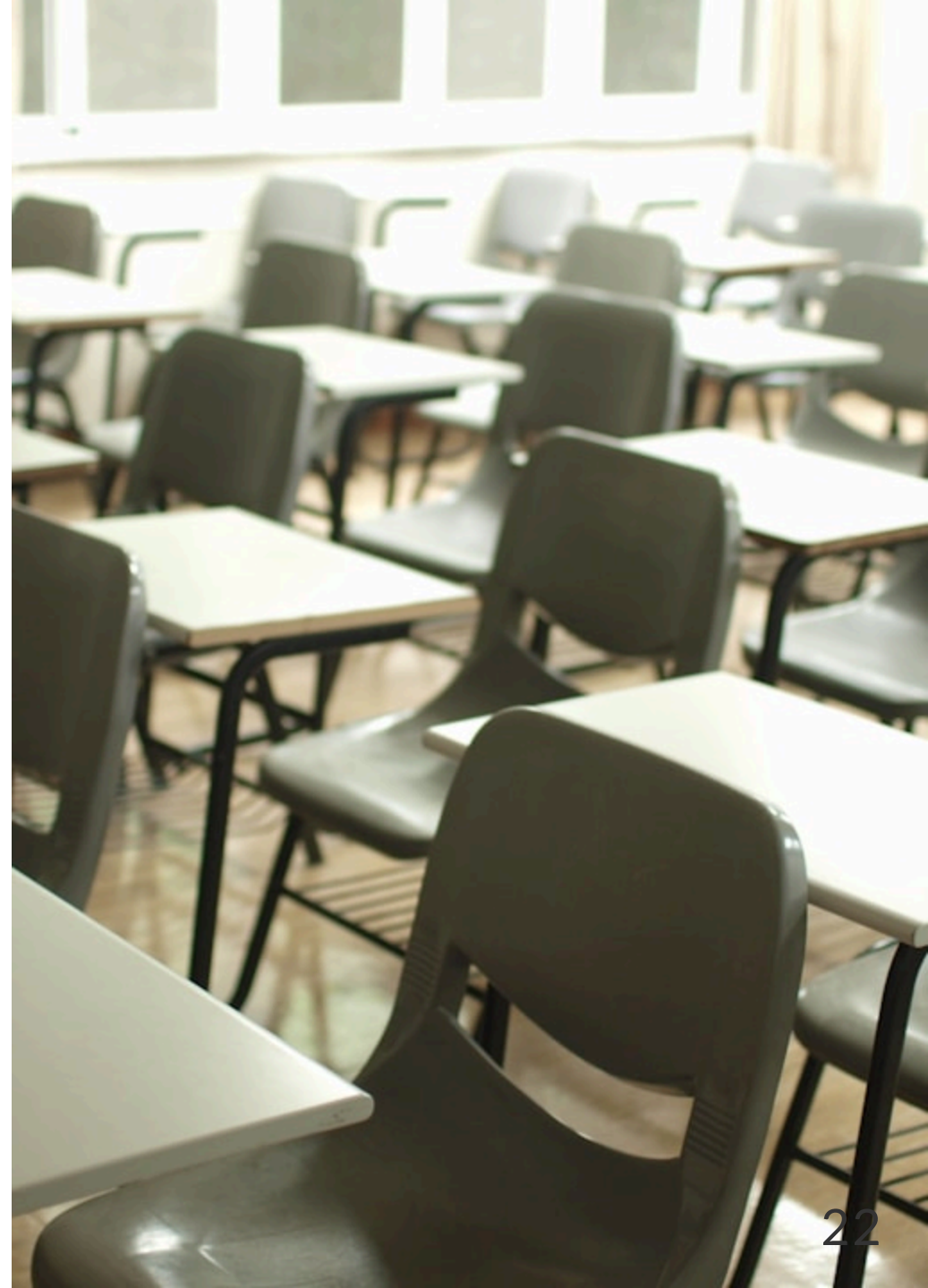
You can use reactions to express your opinion on a comment!

What will you do next?

We are arriving at the end of the second part of the course.

An evaluation will be done to check your understanding of all the content seen in this second part.

More details will be given in the next chapter.



Sources

- Main illustration by [Joanna Kosinska](#) on [Unsplash](#)
- Illustration by [Aline de Nadai](#) on [Unsplash](#)
- Illustration by [Nik](#) on [Unsplash](#)
- Illustration by [MChe Lee](#) on [Unsplash](#)