

Email Security

Emilio Coppa

coppa@diag.uniroma1.it

Sapienza University of Rome



Credits

These slides are based on teaching material originally created by:

- Fabrizio D'Amore (damore@diag.uniroma1.it), Sapienza University of Rome

The Cursed Email

THE INTERNET E-MAIL

The Internet e-mail system

- Background:
 - architecture and functioning
 - extensions (MIME)
- Privacy and security risks:
 - tracking
 - phishing
 - spam
- Validation systems: SPF, DKIM, DMARC, and ARC
- Email security: OpenPGP, S/MIME

Background: how it works

PROTOCOLS

Architecture and message fields:

- Internet Mail Architecture - [RFC 5598](#)
- Internet Message Format - [RFC 5322](#)

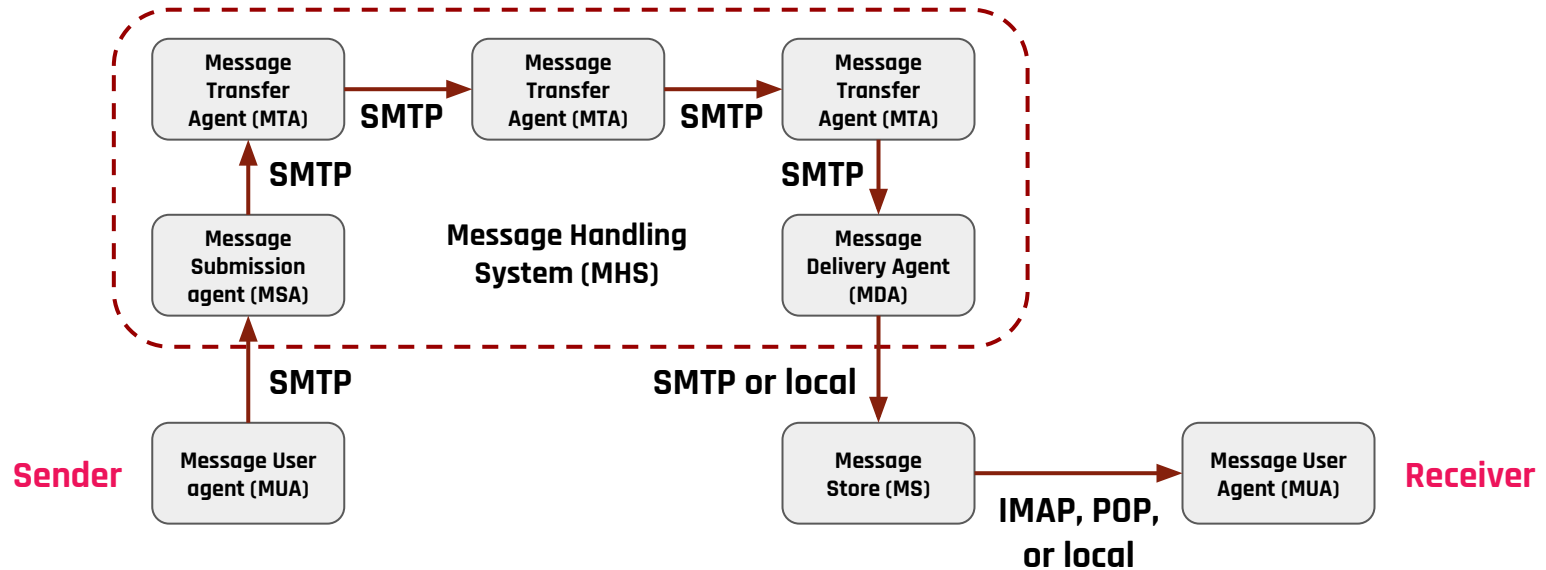
Deliver messages (client-to-server, server-to-server):

- Simple Mail Transfer Protocol (SMTP) - [RFC 780](#) [RFC5321](#)
- Extended Simple Mail Transfer Protocol (ESMTP) - [RFC 1869](#)
- Message Submission - [RFC 2476](#)
- SMTP Service Extension for Authentication (SMTP-AUTH) - [RFC 2254](#)

Retrieve messages (server-to-client):

- Internet Message Access Protocol (IMAP) - [RFC 3501](#)
- Post-Office-Protocol (POP) - [RFC 1939](#)

INTERNET E-MAIL ARCHITECTURE (THEORY)



INTERNET E-MAIL ARCHITECTURE (PRACTICE)

Implementations have often merged different roles into the same software component:

- MSA and MTA
- MTA and MDA

Nowadays, we don't have a MUA but we tend to use a web client (e.g., GMAIL), which often uses proprietary protocol to communicate with the MSA/MTA/MDA.

HOW TO KNOW THE NEXT HOP?

MX record in DNS

```
> dig webhack.it MX
```

```
webhack.it.      300    IN      MX      10 _dc-mx.0f6e60b2ba39.webhack.it.
```

```
> dig 10 _dc-mx.0f6e60b2ba39.webhack.it.
```

```
_dc-mx.0f6e60b2ba39.webhack.it.  209 IN A      62.149.128.151
_dc-mx.0f6e60b2ba39.webhack.it.  209 IN A      62.149.128.163
_dc-mx.0f6e60b2ba39.webhack.it.  209 IN A      62.149.128.157
_dc-mx.0f6e60b2ba39.webhack.it.  209 IN A      62.149.128.160
_dc-mx.0f6e60b2ba39.webhack.it.  209 IN A      62.149.128.166
_dc-mx.0f6e60b2ba39.webhack.it.  209 IN A      62.149.128.154
```

The MX record does not report the port. By default, port 25 is used... Consumer ISP usually blocks this port... hence you cannot host your mail server.

RETRIEVING EMAILS - PROTOCOLS

- POP:
 - supports download and delete operations
 - messages are locally stored using, e.g., a mbox format
 - **messages are fetched by the client and then deleted from the server**
 - connection established only when fetching messages
 - POP3 supports SSL/TLS (default port: 995)
- IMAP:
 - the idea is to permit even different clients to access the same mail box
 - **a copy of the message is left on the server even when fetched by a client**
 - the connection is kept on while the client is browsing the email (faster response time)
 - message state information: read, replied, or deleted
 - server-side searches (which could increase the load...)
 - IMAP supports SSL/TLS (default port: 993)

DELIVERING EMAIL - PROTOCOLS

- SMTP:
 - main commands: **HELO**, **MAIL FROM**, **RCPT TO**, **DATA**, VRFY/EXPN, TURN, **AUTH**, RSET, HELP, QUIT
- ESMTP:
 - new commands: **EHLO**, **STARTTLS**, SIZE, 8BITMIME, ATRN, CHUNKING, DSN, ETRN, PIPELINING, SMTPUTF8, UTF8SMTP
 - support for encrypted connections on SSL/TLS

How to use these commands in practice to send a message?

MESSAGE FORMAT

Three parts:

1. **Message Envelope:**
Like a “shipping label” containing info mainly for routing.
2. **Message Header:**
Metadata like the sender, receiver(s), properties of the message.
3. **Message Body:** the actual content



MESSAGE ENVELOPE

It is designed to contain minimal info about sender/receiver that can be quickly processed by SMTP servers along the route (possibly multi hops). Message headers instead may contain a lot of info, which are mostly useful to the **last** SMTP server.

Main fields:

- **MAIL FROM:** *bounce address*, i.e., where to send back the message in case of failure.
Alternative names: **RETURN PATH, REVERSE PATH, BOUNCE ADDRESS**
- **RCPT TO:** receiver's address, which may be more than one in case of, e.g., CC/BCC.

Example: ssmtp with a ESMTP Server (1)

```
> cat /etc/ssmtp/ssmtp.conf
```

```
FromLineOverride=YES
```

```
hostname=webhack.it           // my domain
```

```
root=noreply@webhack.it
```

```
mailhub=smtps.aruba.it:465    // ESMTP server that will accept ($$$) to deliver mails for my domain
```

```
AuthUser=noreply@webhack.it // username for authentication
```

```
AuthPass=<password>          // password for authentication
```

```
AuthMethod=LOGIN             // username and password will be sent in BASE64
```

```
UseTLS=YES                   // the connection is encrypted with TLS
```

Example: ssmtp with a ESMTP Server (2)

```
> ssmtp -v coppa@diag.uniroma1.it < mail.txt
```

```
[<-] 220 smtpdh15.ad.aruba.it Aruba Outgoing Smtplib ESMTP server ready
```

```
[->] EHLO webhack.it
```

```
[<-] 250 OK
```

```
[->] AUTH LOGIN
```

```
[<-] 334 VXNlcm5hbWU6 // this is "Username:" in base64
```

```
[->] <base64-encoded-username>
```

```
[<-] 334 UGFzc3dvcmQ6 // this is "Password:" in base64
```

```
[->] <base64-encoded-password>
```

```
[<-] 235 2.7.0 ... authentication succeeded
```

```
[->] MAIL FROM:<noreply@webhack.it>
```

```
[<-] 250 2.1.0 <noreply@webhack.it> sender ok
```

```
[->] RCPT TO:<coppa@diag.uniroma1.it>
```

```
[<-] 250 2.1.5 <coppa@diag.uniroma1.it> recipient ok
```

```
[->] DATA // We ask permission to send the other parts of the message
```

```
[<-] 354 OK // Now, we should send the message header and body
```

NOTE: You may use telnet to send these commands. However, most SMTP servers nowadays requires a SSL connection hence it would not work. One solution is to use OpenSSL.

Example: manual session

```
> openssl s_client -connect smtps.aruba.it:465 -crlf -ign_eof
220 smtpdh06.ad.aruba.it Aruba Outgoing Smtplib ESMTPL server ready
> EHLO aasass.com // fake...
250-smtpdh06.ad.aruba.it hello [5.171.189.7], pleased to meet you
> AUTH PLAIN XXXXXXXXXXXX== // echo -ne '\00username\00password' | base64
235 2.7.0 ... authentication succeeded
> MAIL FROM: test@test.com // spoofed address...
250 2.1.0 <test@test.com> sender ok
> RCPT TO: coppa@diag.uniroma1.it
250 2.1.5 <coppa@diag.uniroma1.it> recipient ok
> DATA
354 OK
```


MESSAGE HEADERS

Fields:

- **From:** The email address, and optionally the name of the author(s)
- **Date:** The local time and date when the message was written
- **To:** The email address(es), and optionally name(s) of the message's recipient(s). Indicates primary recipients (multiple allowed)
- **Subject:** A brief summary of the topic of the message. Certain abbreviations are commonly used in the subject, including "RE:" and "FW:"

INFO MESSAGE HEADERS != INFO MESSAGE ENVELOPE

- **TO:** vs **RCPT TO:**

They could be different!

- **FROM:** vs **MAIL FROM:**

They could be different!

Why?

A [common answer](#) is to support automated processes sending mail (e.g., mailing lists) and to send the same message to different receivers (e.g., cc).

MESSAGE HEADERS (2)

Other common fields:

- **Cc:** Carbon copy; Many email clients will mark email in your inbox differently depending on whether you are in the To: or Cc: list.
- **Bcc:** Blind Carbon Copy; addresses added to the SMTP delivery list but not (usually) listed in the message data, remaining invisible to other recipients.
- **Content-Type:** Information about how the message is to be displayed, e.g., a MIME type.
- **Content-transfer-encoding:** encoding used by the content

MESSAGE HEADERS (3)

Other common fields:

- **Message-ID:** Automatically generated field; used to prevent multiple delivery and for reference in In-Reply-To:
- **References:** Message-ID of the message that this is a reply to, and the message-id of the message the previous reply was a reply to, etc.
- **Reply-To:** Address that should be used to reply to the message
- **In-Reply-To:** Message-ID of the message that this is a reply to. Used to link related messages together. This field only applies for reply messages
- **Sender:** Address of the actual sender acting on behalf of the author listed in the From: field (secretary, list manager, etc.)
- **Archived-At:** A direct link to the archived form of an individual email message

TRACE FIELDS IN MESSAGE HEADERS

- **Received:** when an SMTP server accepts a message it inserts this trace record at the top of the header (last to first)
- **Return-Path:** when the delivery SMTP server makes the final delivery of a message, it inserts this field at the top of the header
- **Auto-Submitted:** is used to mark automatically generated messages
- For security checks, other fields may be inserted:
e.g., **Authentication-Results** or **Received-SPF**
we discuss them later on

Example: ssmtp with a ESMTP Server (3)

```
> ssmtp -v coppa@diag.uniroma1.it < Downloads/mail.txt
```

```
...
```

```
[->] DATA
```

```
[<-] 354 OK
```

```
[->] Received: by webhack.it (sSMTP sendmail emulation); Mon, 02 Aug 2021 14:51:59 +0200
```

```
[->] Date: Mon, 02 Aug 2021 14:51:59 +0200
```

```
[->] Bcc: noreply@webhack.it
```

```
[->] From: noreply@webhack.it
```

```
[->] Subject: This is an email
```

```
[->]
```

```
// double \r\n to separate the header and the body
```

```
[->]
```

```
[->] AAAA
```

```
[->] . // \r\n.\r\n to notify end of the DATA
```

```
[<-] 250 2.0.0 AXQFmbklSrXl6AXQGmdGvl mail accepted for delivery
```

```
[->] QUIT
```

```
[<-] 221 2.0.0 smtpdh01.ad.aruba.it Aruba Outgoing Smtip closing connection
```

Example: manual session (2)

```
> openssl s_client -connect smtps.aruba.it:465 -crlf -ign_eof
220 smtpdh06.ad.aruba.it Aruba Outgoing Smtplib ESMTPE server ready
> EHLO aasass.com
250-smtpdh06.ad.aruba.it hello [5.171.189.7], pleased to meet you
> AUTH PLAIN XXXXXXXXXXXX==
235 2.7.0 ... authentication succeeded
> MAIL FROM: test@test.com // spoofed address...
250 2.1.0 <test@test.com> sender ok
> RCPT TO: coppa@diag.uniroma1.it
250 2.1.5 <coppa@diag.uniroma1.it> recipient ok
> DATA
354 OK
> FROM: prova@prova.com // spoofed address...
>
> A\nB\nC
> .
250 2.0.0 KhX3m1L40xC2nKha6mmKAG mail accepted for delivery
```

Result in...

FROM: is shown as the sender

NO WARNING TO ALERT ME THAT COULD BE SPOOFED



prova@prova.com via aruba.it

to ▼

A

B

C

3:44 PM (10 minutes ago)

There are still some open mail relays (1)

John Gilmore (activist)

From Wikipedia, the free encyclopedia

John Gilmore (born 1955) is one of the founders of the [Electronic Frontier Foundation](#), the [Cypherpunks](#) mailing list, and [Cygnus Solutions](#). He created the [alt.* hierarchy](#) in Usenet and is a major contributor to the [GNU Project](#).

An outspoken [civil libertarian](#), Gilmore has [sued](#) the [Federal Aviation Administration](#), [Department of Justice](#), and others. He was the plaintiff in the prominent case [Gilmore v. Gonzales](#), challenging [secret travel-restriction laws](#). He is also an advocate for [drug policy reform](#).

He co-authored the [Bootstrap Protocol](#) in 1985, which evolved into [Dynamic Host Configuration Protocol](#) (DHCP), the primary way local networks assign an [IP address](#) to devices.

Contents [hide]

- 1 Life and career
- 2 Activism
- 3 Affiliations
- 4 Honours
- 5 References
- 6 External links

Life and career [edit]

As the fifth employee of [Sun Microsystems](#) and founder of [Cygnus Support](#), he became wealthy enough to retire early and pursue other interests.

He is a frequent contributor to [free software](#), and worked on several [GNU](#) projects, including maintaining the [GNU Debugger](#) in the early 1990s, initiating [GNU Radio](#) in 1998, starting [Gnash](#) media player in December 2005 to create a free software player for [Flash](#) movies, and writing the [pdtar](#) program which became [GNU tar](#). Outside of the GNU project he founded the [FreeS/WAN](#) project, an implementation of [IPsec](#), to promote the encryption of Internet traffic. He sponsored the [EFF's Deep Crack](#) DES cracker, sponsored the [Micropolis](#) city building game based on [SimCity](#), and is a proponent of [opportunistic encryption](#).

Gilmore co-authored the [Bootstrap Protocol](#) (RFC 951) with Bill Croft in 1985. The Bootstrap Protocol evolved into [DHCP](#), the method by which Ethernet and wireless networks typically assign devices an [IP address](#).

John Gilmore



Gilmore in 2018

Born	1955 (age 65–66) ^{[1][2]} <div>York, Pennsylvania, U.S.</div>
Nationality	American
Known for	Co-Founder of the EFF
Website	www.toad.com/gnu/ 

There are still some open mail relays (2)

Gilmore owns the domain name toad.com, which is one of the 100 oldest active .com domains. It was registered on August 18, 1987. He runs^[when?] the mail server at toad.com as an open mail relay. In October 2002, Gilmore's ISP, Verio, cut off his Internet access for running an open relay, a violation of Verio's terms of service. Many people contend that open relays make it too easy to send spam. Gilmore protests that his mail server was programmed to be essentially useless to spammers and other senders of mass email and he argues that Verio's actions constitute censorship. He also notes that his configuration makes it easier for friends who travel to send email, although his critics counter that there are other mechanisms to accommodate people wanting to send email while traveling. The measures Gilmore took to make his server useless to spammers may or may not have helped, considering that in 2002, at least one mass-mailing worm that propagated through open relays — W32.Yaha — had been hard-coded to relay through the toad.com mail server.^[3]

There are still some open mail relays (3)

\$ telnet new.toad.com 25

Trying 75.101.100.43...

Connected to new.toad.com.

Escape character is '^]'.

HELO sheldoncooper.com

220 hop.toad.com SMTP Sendmail 8.12.9/8.12.9; Sat, 25 Sep 2021 07:21:34 -0700

250 hop.toad.com Hello ppp-251-240.28-151.wind.it [151.28.240.251] (may be forged), pleased to meet you

MAIL FROM: sheldoncooper@tbbt.com

250 2.1.0 sheldoncooper@tbbt.com... Sender ok

RCPT TO: coppa@diag.uniroma1.it

250 2.1.5 coppa@diag.uniroma1.it... Recipient ok

DATA

354 Enter mail, end with "." on a line by itself

FROM: sheldoncooper@tbbt.com

SUBJECT: BAZINGA!!!

Hi mate, BAZINGA!

.

250 2.0.0 18PELY9S017497 Message accepted for delivery

221 2.0.0 hop.toad.com closing connection

Connection closed by foreign host.

There are still some open mail relays (4)



sheldoncooper@tbbt.com



Hi mate, BAZINGA!

16:23 (0 minuti fa)



There are still some open mail relays (5)

Messaggio originale

ID messaggio	<202109251421.18PELY9S017497@hop.toad.com>
Creato alle:	25 settembre 2021 16:21 (consegnato dopo 106 secondi)
Da:	sheldoncooper@tbbt.com
A:	
Oggetto:	BAZINGA!!!
SPF:	NEUTRAL con l'IP 75.101.100.43 Ulteriori informazioni

There are still some open mail relays (6)

```
Delivered-To: coppa@diag.uniroma1.it
Received: by 2002:a05:6918:13cd:b0:5f:cde4:880c with SMTP id m13csp2871822ysj;
    Sat, 25 Sep 2021 07:23:20 -0700 (PDT)
X-Google-Smtp-Source: ABdhPJyFtzX9GH/VqPxBibautnNEyWRo0VzCcc5nDgKfHA6058qfjNrQithjRJvoPcTLj6ZJ8PJ+
X-Received: by 2002:a17:90a:de0f:: with SMTP id m15mr8770808pjv.114.1632579800082;
    Sat, 25 Sep 2021 07:23:20 -0700 (PDT)
ARC-Seal: i=1; a=rsa-sha256; t=1632579800; cv=none;
    d=google.com; s=arc-20160816;
    b=C5SbwKG0lk+8fSR3ruOvtXv4yEsKz184AmY0wJX1MBR5+mooEoXP685jvAhKP/wikB
    /XseTyWfIJZfK9cnnTTKkyZ14ni2Pl12c951mBiCgm0G0IRntdPn5RwDJaGBelk9X3mc
    v0Vnr7o5ZdzRcMfpTqBvG0Feoaws5dYm976musFbSL9/izTPK0oJesljmJVPQoPhVlrB
    Qf5kNLk+SEUPW6Mo3G0oeghpqzedzVZzmsRmKtb5Zo7loFk9+xxx+D2yy6ei815g/Zgfp
    6hadTpZ7ajjZStxlyNynGqbeyQUmZzHDEu0vUz/AkmlDaC3fNygiygeR/6p54xSK9l2H
    WfAQ==
ARC-Message-Signature: i=1; a=rsa-sha256; c=relaxed/relaxed; d=google.com; s=arc-20160816;
    h=subject:from:message-id:date;
    bh=Lor5ls+EtTikLdmW9sZi4wd83zR6pi4yq3Bo6Uq3NT8=;
    b=wAofqGgAo82+E1gYDLq4aSKq4/hUMHNEwx2udGjvTsxGoAEs8KaLi+JzrrUWexDmi
    rKqChUzIoUjEgCz64KvWunz9bXo1loMLL7KeWYuxJ4SREjpM64oXQL9zVRibQL1lmNU3
    eyXwP52vncw5LB4r3kca63z3CoxA43ir7e3t1ldrMmFNwe9p3wQ070Q476ySEZaUc5Ff
    a0A7wNoE+uX2NixJlLtQhlsDFbi7RsfsPTaeAr3lRaGT8VobLXRWnG84xy90ij7ESvd
    BqXmtixbIvlqtC0xLs5J+CYvnRwYyF8eq5vx6qcQwR3Dz/61ZRrc/m970HBH06ZS0zI
    Vvnw==
ARC-Authentication-Results: i=1; mx.google.com;
    spf=neutral (google.com: 75.101.100.43 is neither permitted nor denied by best guess record for domain of sheldoncooper@tbbt.com)
    smtp.mailfrom=sheldoncooper@tbbt.com
    Return-Path: <sheldoncooper@tbbt.com>
Received: from hop.toad.com (75-101-100-43.dsl.static.fusionbroadband.com. [75.101.100.43])
    by mx.google.com with ESMTPS id w190sil3736340pfw.171.2021.09.25.07.23.18
    for <coppa@diag.uniroma1.it>
    (version=TLS1_2 cipher=ECDHE-ECDSA-AES128-GCM-SHA256 bits=128/128);
    Sat, 25 Sep 2021 07:23:19 -0700 (PDT)
Received-SPF: neutral (google.com: 75.101.100.43 is neither permitted nor denied by best guess record for domain of sheldoncooper@tbbt.com) client-ip=75.101.100.43;
Authentication-Results: mx.google.com;
    spf=neutral (google.com: 75.101.100.43 is neither permitted nor denied by best guess record for domain of sheldoncooper@tbbt.com)
    smtp.mailfrom=sheldoncooper@tbbt.com
Received: from sheldoncooper.com (ppp-251-240.28-151.wind.it [151.28.240.251] (may be forged)) by hop.toad.com (8.12.9/8.12.9) with SMTP id 18PELY9S017497 for
    coppa@diag.uniroma1.it; Sat, 25 Sep 2021 07:21:52 -0700
Date: Sat, 25 Sep 2021 07:21:34 -0700
Message-Id: <202109251421.18PELY9S017497@hop.toad.com>
FROM: sheldoncooper@tbbt.com
SUBJECT: BAZINGA!!!
```

Hi mate, BAZINGA!

Training challenge #01

URL: <https://training01.webhack.it>

NOTE: THE CHALLENGE IS LIVE!
TRY IT TO LEARN!

Description:

WebHackIT is happy to offer you an open email relay. However, the developer is a bit dumb and it may not always work as expected...

Can you still send an email using it to admin@webhack.it?

NOTE: this is the first piece of a set of challenges on email security from the WebHackIT CTF.

To access a challenge, you have to register

<https://play.webhack.it/register>

REGISTRATION TOKEN:

webhackit_2222

User Registration

Registration Token

Registration Token

Name

Name

Surname

Surname

Student ID (Matricola)

Student ID (Matricola)

Mail (type the institutional email address if available!)

Email address (institutional email address if available)

Password

Password

Password (Verification)

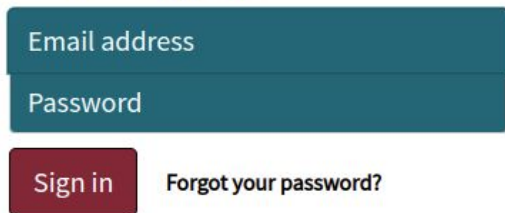
Password (verification)

Register

When opening the challenge, you have to login

- The first time you visit a challenge, e.g., <https://training01.webhack.it>
- You will be redirected to <https://play.webhack.it/login>

Please sign in



The login form consists of two stacked input fields. The top field is labeled 'Email address' and the bottom field is labeled 'Password'. Below these fields is a dark red button labeled 'Sign in'. To the right of the 'Sign in' button is a text link that says 'Forgot your password?'.

- After the login***, visit again the challenge, e.g., <https://training01.webhack.it>
- You should now see the challenge

***** During the login, the portal is setting 2 cookies (“__Host-ctf-platform” and “challenge_auth_token”): the first one is for giving you access to user-specific sections of the CTF portal, the second one is for accessing the challenges (it lasts 30 mins). [DO NOT MESS WITH THESE TWO COOKIES](#). We will discuss cookies later on.**

← → ↻ 🏠 🔒 training01.webhack.it

```
$$$$$$\
$$    $$\
$$ |   $$| $$\   $$\   $$$$ $\ $$$$ \  $$$$ $\ $$$$ \
$$ |   $$| $$|   $$|   $$  $$  $$  $$  $$  $$$$ $\ $$$$ \
$$ |   $$| $$|   $$|   $$  $$  $$  $$  $$  $$$$ $\ $$$$ \
$$ |   $$| $$|   $$|   $$  $$  $$  $$  $$  $$$$ $\ $$$$ \
$$$$$$$ \ $$$$$$$$ \ $$$$ $\ $$$$ \ $$$$ $\ $$$$ \
\      / \      / \      / \      / \      / \      /
```

BY WEBHACK.IT

[->]

```
$$$$$$\  $$\  $$$  $$$$$$$\  $$$$$$$\
$$    $$\  $$$\  $$$  $$$  $$$  $$$  $$$  $$$  $$$  $$$
$$ /    \  |$$$ \  $$$  $$$  $$$  $$$  $$$  $$$  $$$
\$$$$$$\  $$\ $$$  $$$  $$$  $$$  $$$  $$$  $$$  $$$
\    $$$  $$ \ $$$  $$$  $$$  $$$  $$$  $$$  $$$
$$\    $$  |$$  \ $ / $$$  $$$  $$$  $$$  $$$  $$$
\$$$$$$\  $$  \ $ / $$$  $$$  $$$  $$$  $$$  $$$
\      / \      / \      / \      / \      / \      /
```

Analysis

- It is a web application that seems to emulate a TELNET session
- If we type something random, we get back error code that are similar to SMTP errors
- HELP shows some SMTP commands

....let's try to send an email!

```

$$$$$$\
$$ _$$\
$$ |$$| $$\ $$\ $$$$$\$$$$\ $$$$$$$\
$$ |$$| $$\ $$\ $$\ _$$\ _$$\ $$$$$\
$$ |$$| $$\ $$\ $$\ /$$\ /$$\ $$$$$\
$$ |$$| $$\ $$\ $$\ /$$\ /$$\ $$$$$\
$$$$$$$ \$$$$$$$ \$$ \$$ \$$ \$$$$$$$

```

```

$$$$$$\ $$\ $$$ $$$$$$$\ $$$$$$$\
$$ _$$\ $$$\ $$$\ $$$\ $$$\ $$$\
$$ /$$\ $$$\ $$$\ $$$\ $$$\ $$$\
\$$$$$$\ $$$\ $$$\ $$$\ $$$\ $$$\
\ _$$\ $$$\ $$$\ $$$\ $$$\ $$$\
$$$$$$$ $$$\ $$$\ $$$\ $$$\ $$$\
\$$$$$$$ \$$ \$$ \$$ \$$ \$$$$$$$

```

BY WEBHACK.IT

```

[->] HELO webhack.it
[<-] 250 DUMB SMTP SERVER
[->] MAIL FROM: test@test.com
[<-] 250 OK
[->] RCPT TO: admin@webhack.it
[<-] 250 OK
[->] DATA
[<-] 354 End data with <CR><LF>.<CR><LF>
[->] SUBJECT: how are you?
[->]
[->] Hey!
[->] .
[<-]

```

MAIL SUMMARY:
 MAIL FROM: test@test.com
 RCPT TO: [admin@webhack.it]
 DATA: SUBJECT: how are you?

Hey!

FLAG: WIT{ }

[->] █

How to send richer content in the message body?

By default, SMTP supports only **ASCII content**. Exploiting **Content-Type** and **Content-transfer-encoding**, MIME allows to encode the content with:

- BASE64
- QUOTED-PRINTABLE (QP)
- BINARY

Or, when the ESMTP server advertises 8BITMIME, we could directly send 8-bit data. For compatibility reasons, MIME with (BASE64, QP) is often the preferred choice.

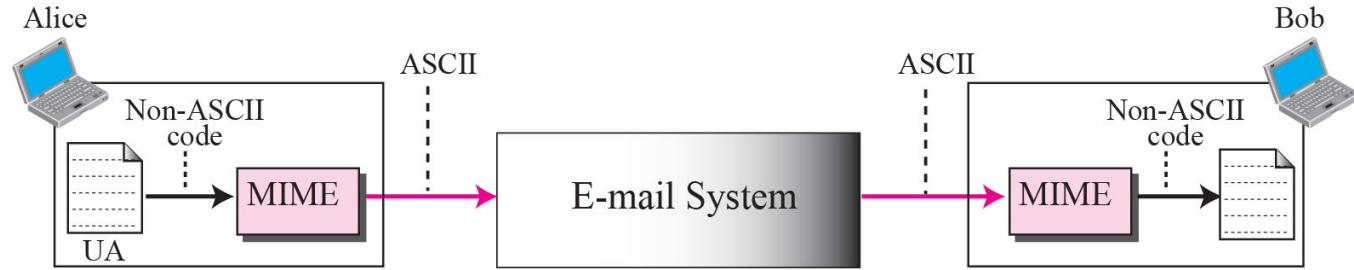
MULTIPURPOSE INTERNET MAIL EXTENSIONS (MIME)

- Text in character sets other than ASCII
- Non-text attachments
- Message bodies with multiple parts
- Header information in non-ASCII character sets

Several RFCs: RFC-822, RFC-2045, RFC-2046, RFC-2047, RFC-2048, RFC-2049

Although MIME was designed for emails, nowadays is used also by other protocols, e.g., HTTP

MIME: main idea



MIME HEADERS

E-mail header
MIME-Version: 1.1 Content-Type: type/subtype Content-Transfer-Encoding: encoding type Content-Id: message id Content-Description: textual explanation of nontextual contents
E-mail body

MIME: type and subtype

Type	Subtype	Description
Text	Plain	Unformatted 7-bit ASCII text; no transformation by MIME is needed
	HTML	HTML format
Multipart	Mixed	Body contains ordered parts of different data types
	Parallel	Body contains no-ordered parts of different data types
	Digest	Body contains ordered parts of different data types, but the default is message/RFC822
	Alternative	Parts are different versions of the same message
Message	RFC822	Body is an encapsulated message
	Partial	Body is a fragment of a bigger message
	External-Body	Body is a reference to another message
Image	JPEG	Image is in JPEG format
	GIF	Image is in GIF format
Video	MPEG	Video is in MPEG format
Audio	Basic	Single channel encoding of voice at 8 KHz
Application	PostScript	Adobe PostScript
	Octet-stream	General binary data (eight-bit bytes)

CONTENT-TRANSFER-ENCODING

Different values:

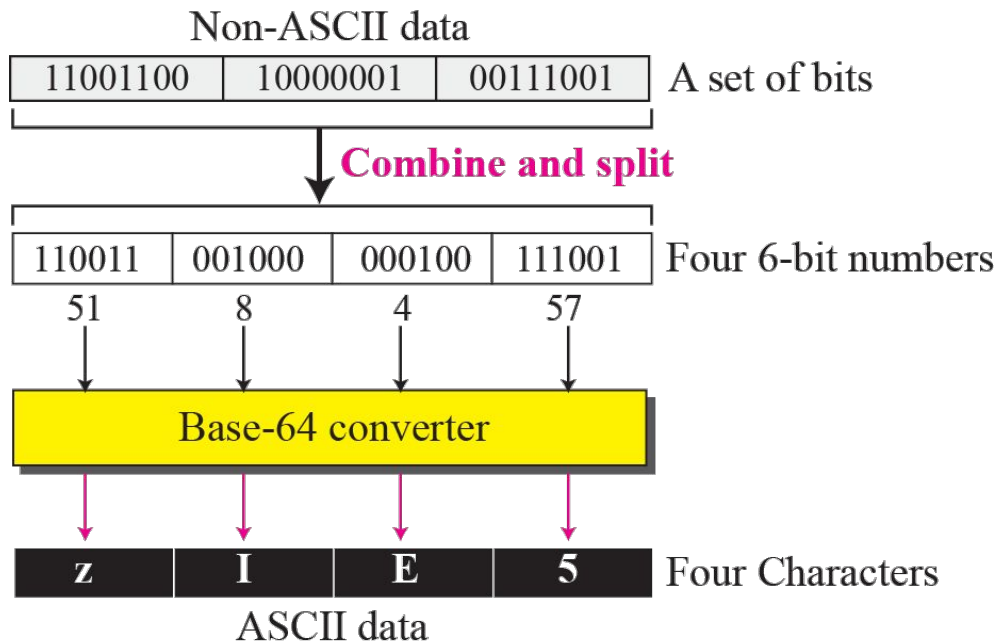
- **7-bit** (ASCII)
- **Binary**
- **Base64**
- **Quoted-Printable**

BASE64

Content-Type: text/plain; charset=ISO-8859-1

Content-transfer-encoding: base64

BASE64 is an encoding scheme that allows to encode any data into ASCII characters. This is done by encoding each block of 6 bits into a 8-bit ASCII values. BASE64 is simple, yet it comes with space overhead (+25% bits).



BASE64 (2)

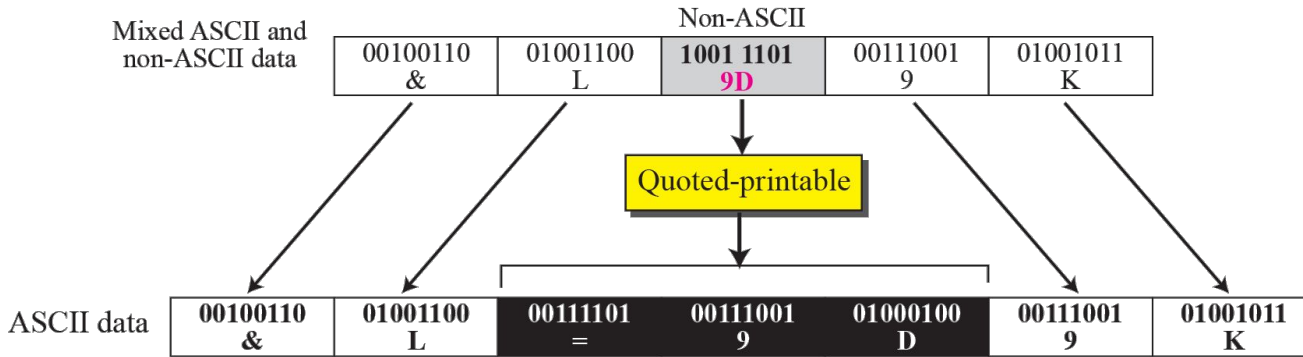
Mapping:

Index	Binary	Char	Index	Binary	Char	Index	Binary	Char	Index	Binary	Char
0	000000	A	16	010000	Q	32	100000	g	48	110000	w
1	000001	B	17	010001	R	33	100001	h	49	110001	x
2	000010	C	18	010010	S	34	100010	i	50	110010	y
3	000011	D	19	010011	T	35	100011	j	51	110011	z
4	000100	E	20	010100	U	36	100100	k	52	110100	0
5	000101	F	21	010101	V	37	100101	l	53	110101	1
6	000110	G	22	010110	W	38	100110	m	54	110110	2
7	000111	H	23	010111	X	39	100111	n	55	110111	3
8	001000	I	24	011000	Y	40	101000	o	56	111000	4
9	001001	J	25	011001	Z	41	101001	p	57	111001	5
10	001010	K	26	011010	a	42	101010	q	58	111010	6
11	001011	L	27	011011	b	43	101011	r	59	111011	7
12	001100	M	28	011100	c	44	101100	s	60	111100	8
13	001101	N	29	011101	d	45	101101	t	61	111101	9
14	001110	O	30	011110	e	46	101110	u	62	111110	+
15	001111	P	31	011111	f	47	101111	v	63	111111	/
Padding		=									

BASE64: an example

Source	Text (ASCII)	M								a								n							
	Octets	77 (0x4d)								97 (0x61)								110 (0x6e)							
Bits		0	1	0	0	1	1	0	1	0	1	1	0	0	0	0	1	0	1	1	0	1	1	1	0
Base64 encoded	Sextets	19								22								5							
	Character	T								W								F							
	Octets	84 (0x54)								87 (0x57)								70 (0x46)							

QUOTED-PRINTABLE ENCODING



- any 8-bit byte value may be encoded with 3 characters: an '=' followed by two hexadecimal digits (0-9 or A-F) representing the byte's numeric value
- non 8-bit byte values are ASCII chars from 33 to 126 (excluded 61, the '=' sign)
- special cases for SPACE and TAB
- **more space efficient than BASE64 but it not *space uniform***

MULTIPART SUBTYPES

- **Mixed.** For sending files with different "Content-Type" headers.
- **Digest.** To send multiple text messages.
- **Message.** Contains any MIME email message, including any headers
- **Alternative.** Each part is an "alternative" version of the same (or similar) content (e.g., text + HTML)
- more subtypes...

MULTIPART/MIXED: EXAMPLE

From: Some One <someone@example.com>

MIME-Version: 1.0

Content-Type: multipart/mixed; boundary="XXXXboundary text"

This is a multipart message in MIME format.

A client that does not fully support MIME, will show this text.

--XXXXboundary text

Content-Type: text/plain

this is the body text

A client that does support MIME, will show this text.

--XXXXboundary text

Content-Type: text/plain;

Content-Disposition: attachment; filename="test.txt"

this is the attachment text

A client that does support MIME, show this an attachment within the UI.

--XXXXboundary text--

MULTIPART/ALTERNATIVE: EXAMPLE

MIME-Version: 1.0

Content-Type: multipart/alternative; boundary="-----=_Part_804988_864270330.1627985944031"

-----=_Part_804988_864270330.1627985944031

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

Content-Transfer-Encoding: quoted-printable

Content-ID: text-body

Dear Emilio, [...]

-----=_Part_804988_864270330.1627985944031

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

Content-Transfer-Encoding: quoted-printable

Content-ID: html-body

<html>[...]Dear Emilio,
[...]</html>

-----=_Part_804988_864270330.1627985944031--