Електронна поща

Общи положения

Електронната поща (**e-mail** или **email)** е метод за обмен на електронни съобщения.

Едно съобщение се състои най-малко от съдържанието си, адрес на автора и адресите на един или повече получатели.

Корените на днешната email са в Arpanet – стандарт за кодиране на съобщения - RFC 733 (подновена с 822... 5322).

Преходът от Arpanet към Internet в началото на 1980-те добави постепенно новостите към основната услуга:

- транспортния протокол Simple Mail Transfer Protocol (SMTP), RFC 821 през 1982 г. (обновена с 2821... 5321)

- прикрепяния на мултимедия – от 1996 г. - от RFC 2045 до RFC 2049, известни като Multipurpose Internet Mail Extensions (MIME).

Общи положения

Email се базират на модела с пълно буфериране (store-and-forward).

Сървърът за електронна поща приема, препраща, доставя или съхранява съобщения за сметка на потребителите.

Тяхна задача е само да се свържат към email инфраструктурата с помощта на компютрите си.

Терминология

Mail-box – файл или директория/и от файлове, където се съхраняват входящите съобщения.

mail user agent (MUA) е приложна програма, стартирана от потребителя. Използва се за оформяне и изпращане на съобщения, както и за показване, сортиране като файлове и принтиране на получени в кутията съобщения. Такива са **Mozilla Thunderbird**, MS Outlook и др.

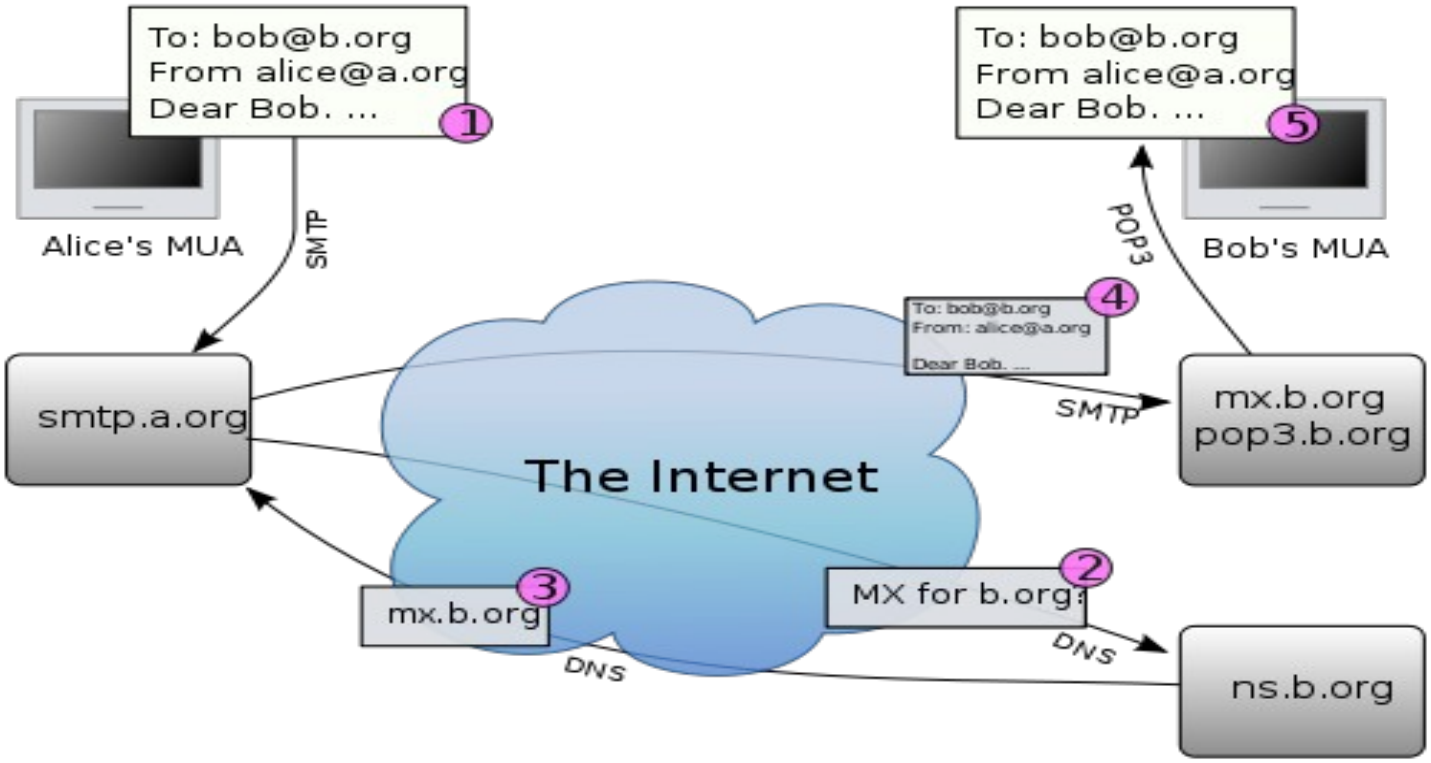
Терминология

Mail transfer agent (MTA) осъществява маршрутизацията на съобщенията, подадени от MUA, до получателя.

Преносът на пощенски съобщения се дефинира от протокола SMTP.

Delivery agent поставя съобщението в пощенската кутия на потребителя. Доставянето се дефинира от протоколите POP3 или IMAP4.

Alice си пише с Bob



Какво става

1. mail user agent (MUA) на Алис форматира съобщението в e-mail формат и с помощта на SMTP го изпраща към местния mail transfer agent (MTA) - smtp.a.org.
2. MTA гледа за крайния адрес, според SMTP протокола (а не в главата на съобщението) - bob@b.org. В e-mail адреса частта пред @ е локалната част, най-често потребителското име на получателя. Частта след @ е име на домейна. MTA по името на домейна определя пълното домейн име на пощенския сървър в DNS.
3. DNS сървърът за домейн b.org - ns.b.org, отговаря с MX записи, изброяващи пощенските сървъри в този домейн, в случая mx.b.org.
4. smtp.a.org изпраща съобщението до mx.b.org по SMTP, който го доставя до пощенската кутия (mailbox) на bob.
5. Bob натиска бутон "get mail" на своя MUA, с което изтегля съобщението с помощта на (в случая) Post Office Protocol (POP3).

Алтернативи на последователността

- Alice може да няма MUA на компютъра си, а да се свърже към webmail услуга.
- На компютъра на Alice може да е “качен” MTA, т.е да прескочи стъпка 1.
- Bob има много начини да изтегли пощата си, например, по протокол Internet Message Access Protocol, като се логне на mx.b.org и си я чете директно от там, или и той с webmail.
- В един домейн има няколко пощенски сървъра, така че те могат да продължат да приемат поща, даже когато главният е отпаднал.
- За повишаване на сигурността и конфиденциалността е добре пощата да се криптира (OpenPGP, X.500 сертификати). Но това е друга тема.

Алгоритъм на MX йерархия в DNS.

...

$ORIGIN domain.uni-sofia.bg.

MX 10 mail.*faculty*.uni-sofia.bg

MX 20 ns.uni-sofia.bg.

MX 30 ady.uni-sofia.bg.

MX йерархия

Съгласно горната MX йерархия при иницииране на сесия за предаване на писмо към получател с пощенска кутия в домейна *faculty*.uni-sofia.bg:

- опит да се установи SMTP сесия към mail.*faculty*.uni-sofia.bg. Ако този опит пропадне:

- SMTP сесия към SMTP сървъра с MX приоритет 20 (ns.uni-sofia.bg) или

- с MX приоритет 30 (ady.uni-sofia.bg).

open mail relay

MTAs, които приемат съобщения от произволни податели и полагат максимални усилия да ги препратят по посока към получателите.

Такива MTAs се наричат open mail relay.

В зората на Internet, когато мрежите не бяха надеждни, това усилие беше похвално. Да може съобщението все пак да достигне целта си през един или повече relay.

Но от този механизъм се възползваха недобросъвестни “изпращачи” на спам и друга нерегламентирана поща.

Затова днешните MTAs не са open mail relays и не приемат поща от open mail relays, която си чист спам.

Това важи и за сървърите СУ. Използваме и отворен софтуер SpamAssassin.

open relay denied (пример)

```
[stefan@shuttle ~]$ telnet email.uni-sofia.bg 25
Trying 62.44.101.22...
Connected to email.uni-sofia.bg (62.44.101.22).
Escape character is '^]'.
220 email.uni-sofia.bg ESMTP Postfix
HELO email.uni-sofia.bg
250 email.uni-sofia.bg
MAIL FROM:alabala@gmail.com
250 2.1.0 Ok
RCPT TO:abracadabra@yahoo.com
554 5.7.1 <abracadabra@yahoo.com>: Relay access denied
QUIT
221 2.0.0 Bye
```

Формати

Форматът на e-mail съобщенията е дефиниран в RFC 5322 и серия от RFC-та, RFC 2045 до RFC 2049, "Multipurpose Internet Mail Extensions" или MIME.

e-mail съобщенията се състоят от два основни дяла, отделени с празен ред:

Header (глава) Структурирано е от полета, обобщение (summary), подател (sender), получател (receiver) и др.

Body (тяло) Самото съобщение като неструктуриран текст. Понякога завършва и с “подпис”, signature block.

Полета в основното заглавие

Заглавието включва най-малко следните полета:

From: e-mail address и евентуално името на изпращача. При подателя се попълва автоматично.

To: Адрес(ите) и евентуално име(ната) на получател(ите).

CC: До кой да се изпрати видимо за получателя To: копие.

Bcc: Blind Carbon Copy До кой да се изпрати невидимо за получателя To: копие.

Subject: Или Относно: Предмета на съобщението.

Date: Дата и час на изпращане в локалното време. Поставя се автоматично.

Спекулации с "From"

С полето "From" може лесно да се заблуждава, затова се препоръчва да се ползва цифрово подписване (OpenPGP или X.500 сертификат).

Други важни полета в разширеното заглавие

In-Reply-To: Message-ID на съобщението, на което настоящото е отговор.

Received: Проследява пътя, по който е минало съобщението, през кои пощенски сървъри. Показва кой е истинския подател по IP адрес.

References: Message-ID на това съобщение и на това, на което е отговор.

Reply-To: Адресът за отговор на подателя.

Пример. Заглавие на phishing съобщение

Subject: Уважаеми Uni-sofia.bg потребителски акаунт

From: Софийски университет <web-master@Uni-sofia.bg>

Date: Sun, March 21, 2010 13:22

To: undisclosed-recipients:;

Priority: Normal

Mailer: SquirrelMail/1.4.13

Полета в разширеното заглавие

Return-Path: <web-master@Uni-sofia.bg>

Received: from mailbox.uni-sofia.bg ([unix socket])

by mailbox.uni-sofia.bg (Cyrus v2.3.7-Invoca-RPM-2.3.7-7.el5_4.3) with LMTPA; Sun, 21 Mar 2010 13:22:36 +0200

X-Sieve: CMU Sieve 2.3

Received: from olc-11.verat.net (olc-11.verat.net [62.108.127.37])

by mailbox.uni-sofia.bg (8.13.8/8.13.8) with ESMTP id o2LBMYL9015117

for <stefan@ucc.uni-sofia.bg>; Sun, 21 Mar 2010 13:22:35 +0200

...разширеното заглавие

Received: from webmail.verat.net (webmail.verat.net [85.222.160.153]) by olc-11.verat.net (Postfix) with ESMTP id D595BFC999; Sun, 21 Mar 2010 12:18:43 +0100 (CET)

Received: from 41.138.189.77(SquirrelMail authenticated user djmaxa) by webmail.verat.net with HTTP; Sun, 21 Mar 2010 12:22:33 +0100 (CET)

Message-ID: <3754.41.138.189.77.1269170553.squirrel@webmail.verat.net>

Date: Sun, 21 Mar 2010 12:22:33 +0100 (CET)

...разширеното заглавие

Subject:

=?windows-1251?Q?
=D3=E2=E0=E6=E0=E5=EC=E8_Uni-sofia.bg_

From: =?windows-1251?Q?=D1=EE=F4 ... =?windows-1251?Q?=E5=F2?= <web-master@Uni-sofia.bg>

Reply-To: w0642406@gmail.com

User-Agent: SquirrelMail/1.4.13

MIME-Version: 1.0

Content-Type: text/plain;charset=windows-1251

Content-Transfer-Encoding: 8bit

X-Priority: 3 (Normal)

Importance: Normal

...разширеното заглавие

To: undisclosed-recipients:;

X-Spam-Status: No, score=-2.6 required=5.0 tests=BAYES_00 autolearn=ham

version=3.2.5

X-Spam-Checker-Version: SpamAssassin 3.2.5 (2008-06-10) on mailbox.uni-sofia.bg

Кодиране. UTF-8.

Първоначалносъобщенията в ел. поща са в 7-bit ASCII код.

Стандартът MIME въвежда предаване и на не-ASCII данни.

UTF-8 (8-bit **UCS**[1]/Unicode Transformation Format) дефинира кодиране с променлива дължина на всичките 1,112,064 валидни кодови точки в **Unicode**[2].

Представя всеки знак в Unicode стандарта, но в същевременно е обратно съвместим с ASCII.

Затова става все по-предпочитан за e-mail, web и др.

UTF-8 кодира всеки знак (code point) с 1 до 4 байта, като с един байт се кодират 128-те US-ASCII знаци.

Internet Mail Consortium (IMC) препоръчва всички email програми да са в състояние да изобразяват и създават поща с помощта UTF-8.

[1]Universal Character Set (**UCS**) **ISO/IEC 10646** стандарт, разработен съвместно с Unicode Consortium.

[2]**Unicode** осигурява уникален номер за всеки знак, независимо от платформата, независимо от програмата, независимо от езика.

UTF-8

Първите 128 знака (US-ASCII) им трябва 1 байт.

Следващите 1920 – 2 байта. Това са латински букви с диакрити, гръцки, кирилица, арменски, арабски, иврит и др.

3 байта са необходими за за останалите лингвистични знаци.

4 байта – за знаци в други равнини на Unicode, рядко използвани в практиката.

SMTP

Протоколът за изпращане на поща е **SMTP** (Simple Mail Transfer Protocol) се дефинира в **RFC 5321**.

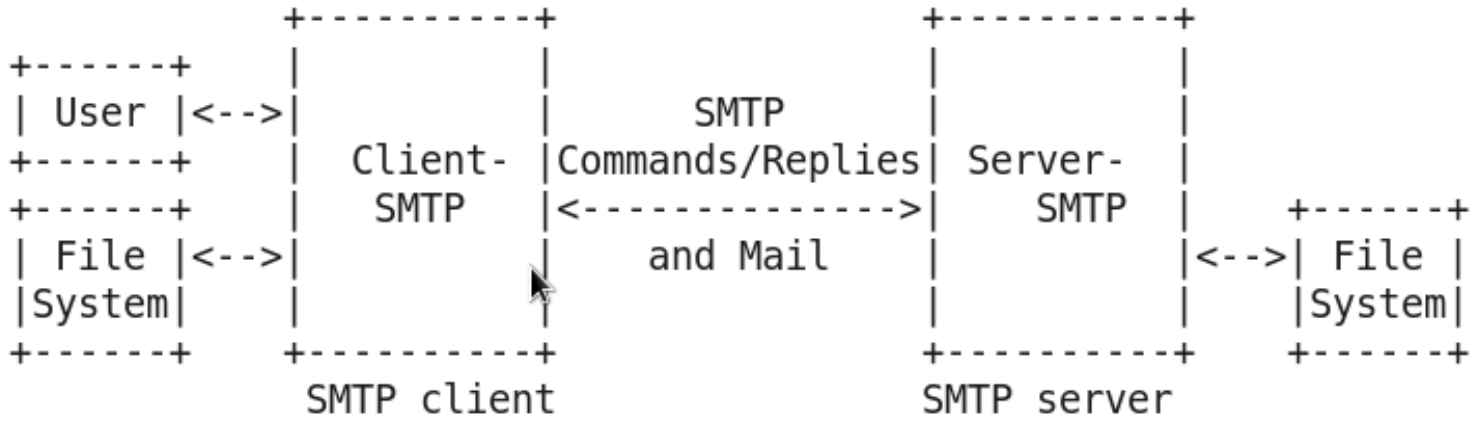
Базира се на транспортен протокол TCP.

От клиента, от порт с номер по-голям от 1024, се прави заявка за сесия към IP адреса на пощенския сървър на порт 25, т.е. порт 25 стои отворен в пощенския сървър и чака заявка за съединение.

Ако сървърът е в състояние да получи заявката, отговаря с 220, което означава готов.

След това клиентът изпраща съобщение HELLO, а при успех сървърът отговаря с 250.

SMTP. Как работи.



SMTP

След това клиентът изпраща MAIL FROM (от кого е пощата), RCPT TO (кой е получателя) и накрая се прехвърля самото съобщение, след което връзката се разпада.

Описаното си е една TCP/IP сесия. В неговите рамки се обменят ASCII съобщения, които са с определена структура.

Минимална реализация

За да работи SMTP, приемниците трябвада реализират минимално следните команди:

EHLO

HELO

MAIL

RCPT

DATA

RSET

NOOP

QUIT

VRFY

Пример на SMTP сесия

По-долу имате един типичен пример на изпращане на съобщение по SMTP до две пощенски кутии (*alice* и *theboss*) в един и същ домейн (*example.com*).

“Репликите” на сървъра са означени със (*S:*), а на клиента - с (*C:*).

След като изпращачът на съобщението (STMP client) установи надежден канал с получателя (SMTP server), сесията се отваря с поздравление от страна на сървъра.

Клиентът започва диалог, отговаряйки с команда HELO, в която се идентифицира.

Пример на SMTP сесия

S: 220 smtp.example.com ESMTP Postfix
C: HELO relay.example.org
S: 250 Hello relay.example.org, I am glad to meet you
C: MAIL FROM:<bob@example.org>
S: 250 Ok
C: RCPT TO:<alice@example.com>
S: 250 Ok
C: RCPT TO:<theboss@example.com>
S: 250 Ok
C: DATA
S: 354 End data with <CR><LF>.<CR><LF>

Пример (подробно от клиента)

C: From: "Bob Example" <bob@example.org>
C: To: Alice Example <alice@example.com>
C: Cc: theboss@example.com
C: Date: Tue, 15 Jan 2008 16:02:43 -0500
C: Subject: Test message
C:
C: Hello Alice,
C: This is a test message with 5 header fields and 4 lines in the message body.
C: Your friend,
C: Bob
C: .

Пример (прод.)

S: 250 Ok: queued as 12345

C: QUIT

S: 221 Bye

{Сървърът затваря сесията}

SMTP продукти с отворен код. Sendmail и Postfix

Sendmail е софтуер за маршрутизация на електронна поща, поддържащ и SMTP. Оригиналът е написан от Eric Allman. От 2013 г. е собственост на Proofpoint, Inc.

sendmail.org отговаря на https://www.proofpoint.com/us/open-source-email-solution

Postfix (www.postfix.org) е mail server, писан от Wietse Venema в IBM като алтернатива на Sendmail. Целта е да е бърз, лесен за администриране и сигурен.

Някои характеристики изискват външни библиотеки (LDAP, SQL, TLS), а други разчитат на заложеното в операционната система

sendmail и IPv6

sendmail е компилиран с поддръжка на IPv6.

За IPv6 интеграция в SMTP се въвеждат следните опции:

```
DAEMON_OPTIONS(`Port=smtp,Addr=62.44.109.37, Name=MTA')dnl
DAEMON_OPTIONS(`Port=smtp,Addr=127.0.0.1, Name=MTA')dnl
DAEMON_OPTIONS(`Port=smtp,Addr=2001:67c:20d0:10::37, Name=MTA6,
Family=inet6')dnl
DAEMON_OPTIONS(`Port=smtp,Addr=::1, Name=MTA6, Family=inet6')dnl
```

Multipurpose Internet Mail Extensions (MIME)

MIME разширява стандартната електронна поща с:

Текст не само в ASCII формат;

Нетекстови приложения (attachments): аудио, видео, изображения, приложни програми и др.;

Съобщения, разделени на множество части

Информация в Header в не-ASCII формат.

MIME стандарти

MIME се дефинира в шест последователни RFC-та: 2045, 2046, 2047, 4289, 2049 и 6838.

Типовете съдържание, дефинирани в MIME стандартите, се прилагат и в други протоколи като HTTP и SIP. Web вмъкват MIME хедър в началото на всеки Web пренос.

Secure/Multipurpose Internet Mail Extensions (S/MIME)

S/MIME (RFC 5751) осигурява следните криптографски услуги:

- аутентикация;
- интегритет на данните в съобщението;
- гарантиране на източника на съобщението (non-repudiation) чрез цифрово подписване;
- и конфиденциалност на данните чрез криптиране.

Освен това, S/MIME предлага и компресиране.

Mail Delivery. POP3.

Крайният получател на писмото не е SMTP-сървъра. В него се събират изпратените писма до съответния домейн.

Сървърът трупа тези писма на диск при себе си.

С помощта на друг протокол крайният получател изтегля получените писма от пощенския сървър.

Например, **POP3** (Post Office Protocol – RFC 1939). При него сървърът слуша на **tcp/110**. Пощата се изтегля от сървъра, на който е оставена (maildrop) след което се изтрива от там.

POP3 поддържа автентикация на клиента (username + password), т.е. притежателят на пощенската кутия е регистриран като POP3 потребител.

Когато клиентът се свърже към POP3 сървъра, той първо се идентифицира, след което може да извърши други команди за прочитане на получените от него mail-ове.

IMAP

Internet Message Access Protocol (**IMAP**) “слуша” на **tcp/143** и е другата възможност крайният клиент да получи достъп до пощата си, стояща на отдалечен сървър.

Сегашната версия, IMAP version 4 revision 1 (IMAP4rev1) е дефинирана в **RFC 3501**.

IMAP поддържа и online, и offline режими.

E-mail клиенти с IMAP оставят съобщенията на сървъра, докато потребителят не реши да ги изтрие.

IMAP позволява повече от един клиент да има достъп до една и съща пощенска кутия. Т.е можете да имате достъп до пощата си едновременно от няколко места.

IMAP

IMAP4 клиентите могат да създават, преименуват и/или изтриват пощенски кутии (потребителят ги вижда като папки) на сървъра и да местят съобщения между кутиите.

POP3 и IMAP услуги предлагат Cyrus IMAP server (http://cyrusimap.web.cmu.edu/) и Dovecot (www.dovecot.org).

IMAP4 команди

IMAP4 Commands

Command	Syntax	Description
AUTHENTICATE	a100 AUTHENTICATE method + challenge response + challenge response	Authenticates on the IMAP4 server via a secure authentication method.
LOGIN	a100 LOGIN username password	Authenitcates on the IMAP4 server via plaintext.
LIST	a102 LIST "" *	Lists contents of an account.
SELECT	a102 SELECT INBOX	Selects a mailbox.
EXAMINE	a102 EXAMINE INBOX	Returns statistics on a mailbox, without selecting it.
CREATE	a104 CREATE mailbox a104 CREATE directory\	Creates a new mailbox or directory hierarchy on the server. Note: "\" should match the hierarchy separator returned by the LIST command.
DELETE	a102 DELETE mailbox	Deletes a mailbox or a directory hierarchy.
RENAME	a102 RENAME mailbox name	Renames a mailbox or a directory hierarchy.

Документация на mail решение

Системата включва:

Sendmail за MTA

Dovecot за MDA

RedHat Directory Server за LDAP база за потребителски акаунти и конфигурация на sendmail

phpLDAPadmin - за външно управление на LDAP директорията

Saslauthd за посредник между dovecot/imap и LDAP базата

Roundcube webmail (https://roundcube.net/), написана на PHP Version >= 5.4

Документация (прод.)

Spamassassin за антиспам защита.
(*Използва шаблони, по които анализира съдържанието на пощата и го класифицира като спам или не. Подлежи на самообучение.*)

Clamav (www.clamav.net) за антивирусна защита.

MIMEDefang (http://www.mimedefang.org/)– средство за филтриране на e-mail. Работи заедно с библиотеката "Milter" на Sendmail. Политиките да се пишат на Perl вместо на C, което ги прави по-бързи.

TLS / SSL

TLS (Transport Layer Security) е протокол за криптирани комуникации на транспортно ниво в мрежата.

Прилага се при комуникациите по e-mail (SMTPS, IMAPS, POP3) и web (HTTPS), по FTP и др. за криптиране на данните.

Последната версия е TLS 1.3 (**RFC 8446**).

Той е развитие на Secure Sockets Layer (SSL), разработен от Netscape през 1990-те.

TLS 1.3 ръкостискане

В TLS 1.3 "ръкостискането" става само на два етапа, а не на 4, както е в предишните версии.

Клиентът стартира с ClientHello и списъка с поддържани шифри. Освен това отгатва кой алгоритъм за обмен на ключове ще избере сървъра, като изпраща споделен ключ.

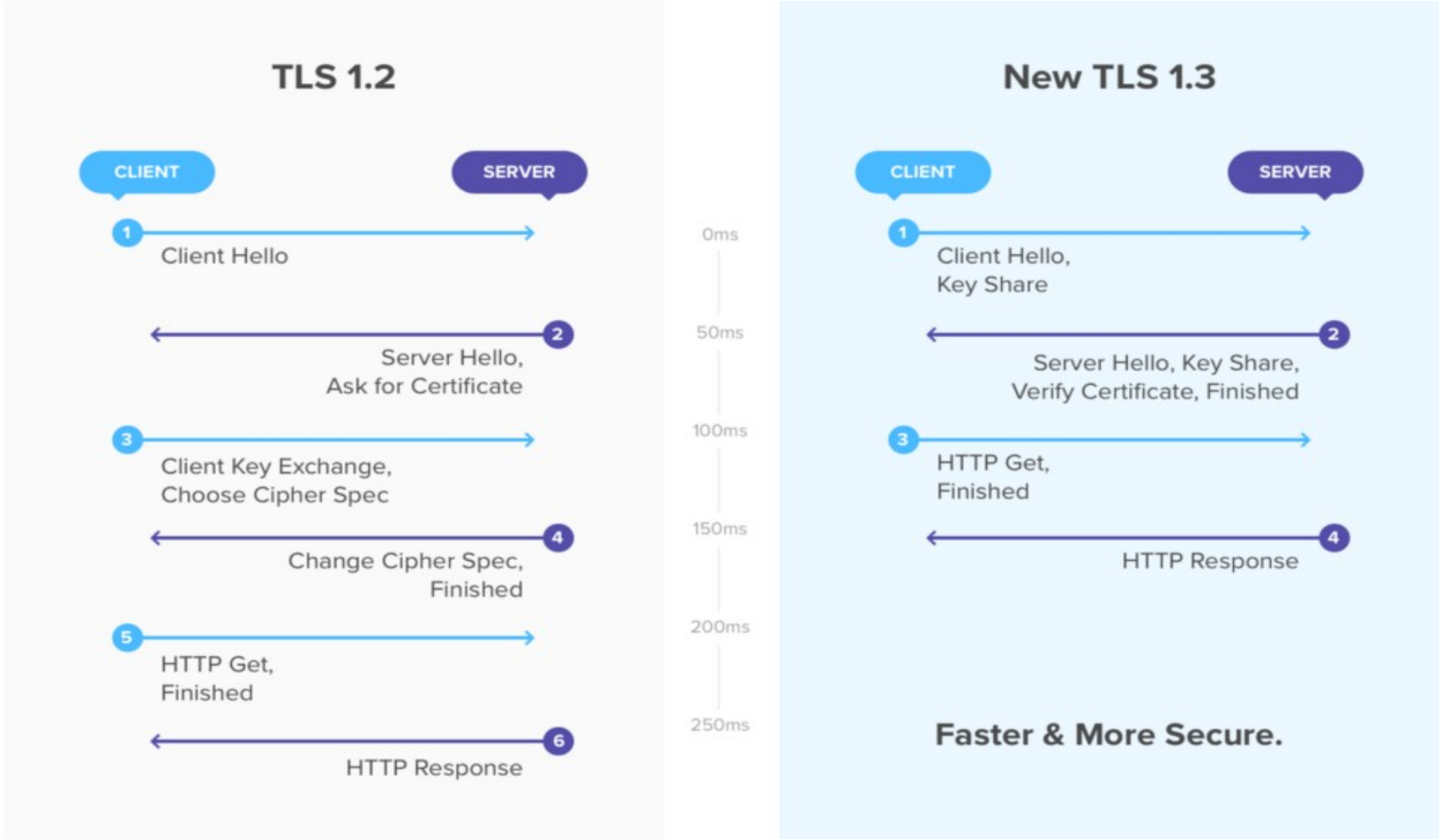
TLS 1.3 ръкостискане (прод.)

Получавайки ClientHello, сървърът избира шифро пакета и алгоритъма за обмен на ключове. С което е готов да генерира ключ и да превключи към пренос на криптирани пакети.

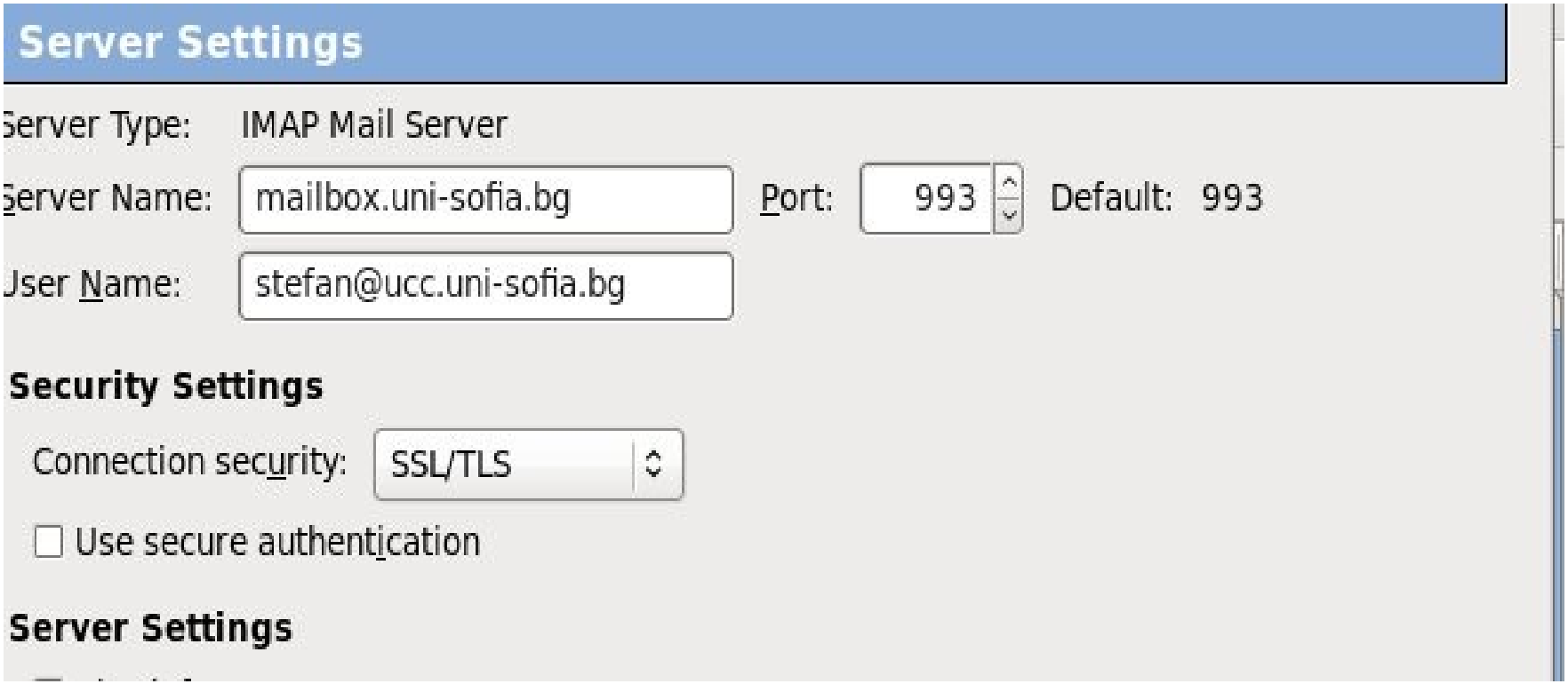
Така сървърът изпраща ServerHello, своя споделен ключ, криптираният с него сертификат и съобщението Finished.

Клиентът генерира ключовете с помощта на споделения, проверява сертификата и Finished, с което е готов да изпрати криптирана заявка, например HTTP request.

TLS 1.2 vs. 1.3


TLS 1.2
CLIENT
SERVER
1
Client Hello
2
Server Hello,
Ask for Certificate
3
Client Key Exchange,
Choose Cipher Spec
4
Change Cipher Spec,
Finished
5
HTTP Get,
Finished
6
HTTP Response
0ms
50ms
100ms
150ms
200ms
250ms
New TLS 1.3
CLIENT
SERVER
1
Client Hello,
Key Share
2
Server Hello, Key Share,
Verify Certificate, Finished
3
HTTP Get,
Finished
4
HTTP Response
Faster & More Secure.

Защитена поща. IMAPS.



Защитена поща. SMTPS.

Защитена поща

Secure SMTP (SMTPS) - port 465

IMAP4 over SSL (IMAPS) - port 993

Паролата и потребителското име са криптирани при преноса им до сървъра.

Служебната информация, отнасяща се до съдържанието на писмата и процедурите на протокола (SMTP / IMAP), се пренася в Интернет в криптиран вид!

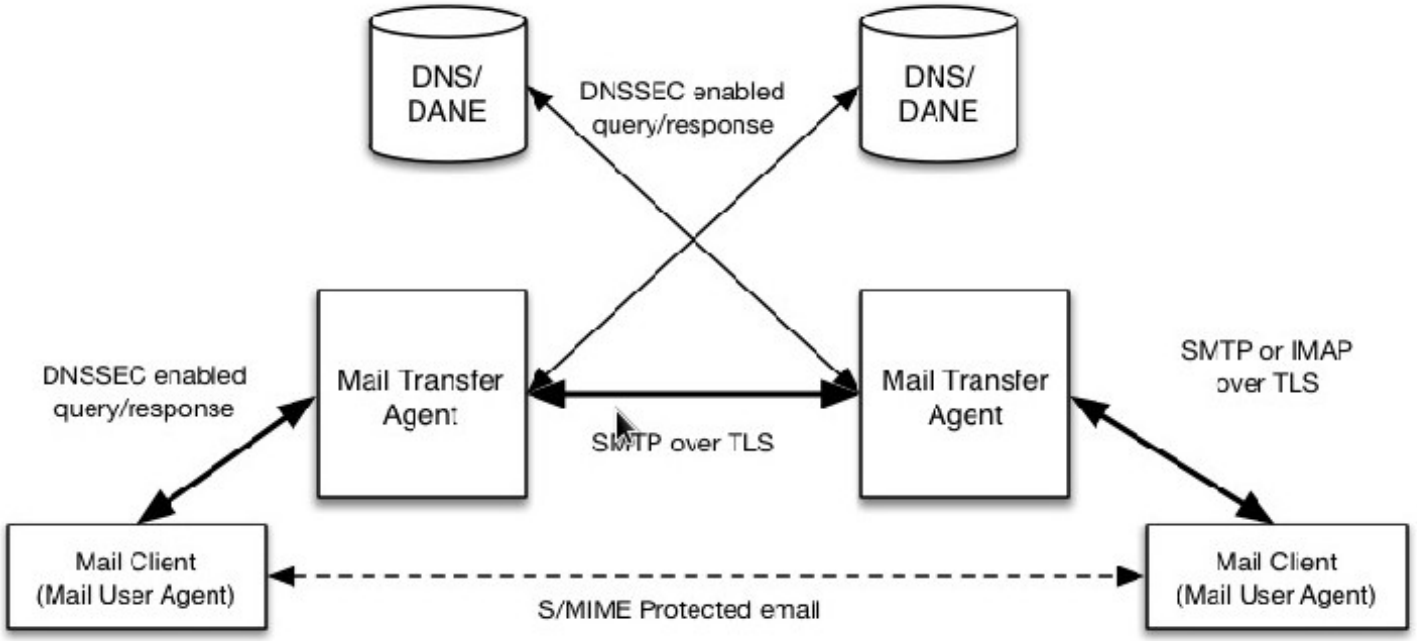
Защитена поща. Решение на NIST.

NIST наскоро публикува Special Publication (SP) 1800-6, Domain Name System-Based Electronic Mail Security.

Този проект включва MUAs, DNS сървъри, MTAs и източници на X.509 криптографски ключове за сертификати.

Фокусира се върху SMTP over TLS и S/MIME.

Защитена поща. Решение на NIST.



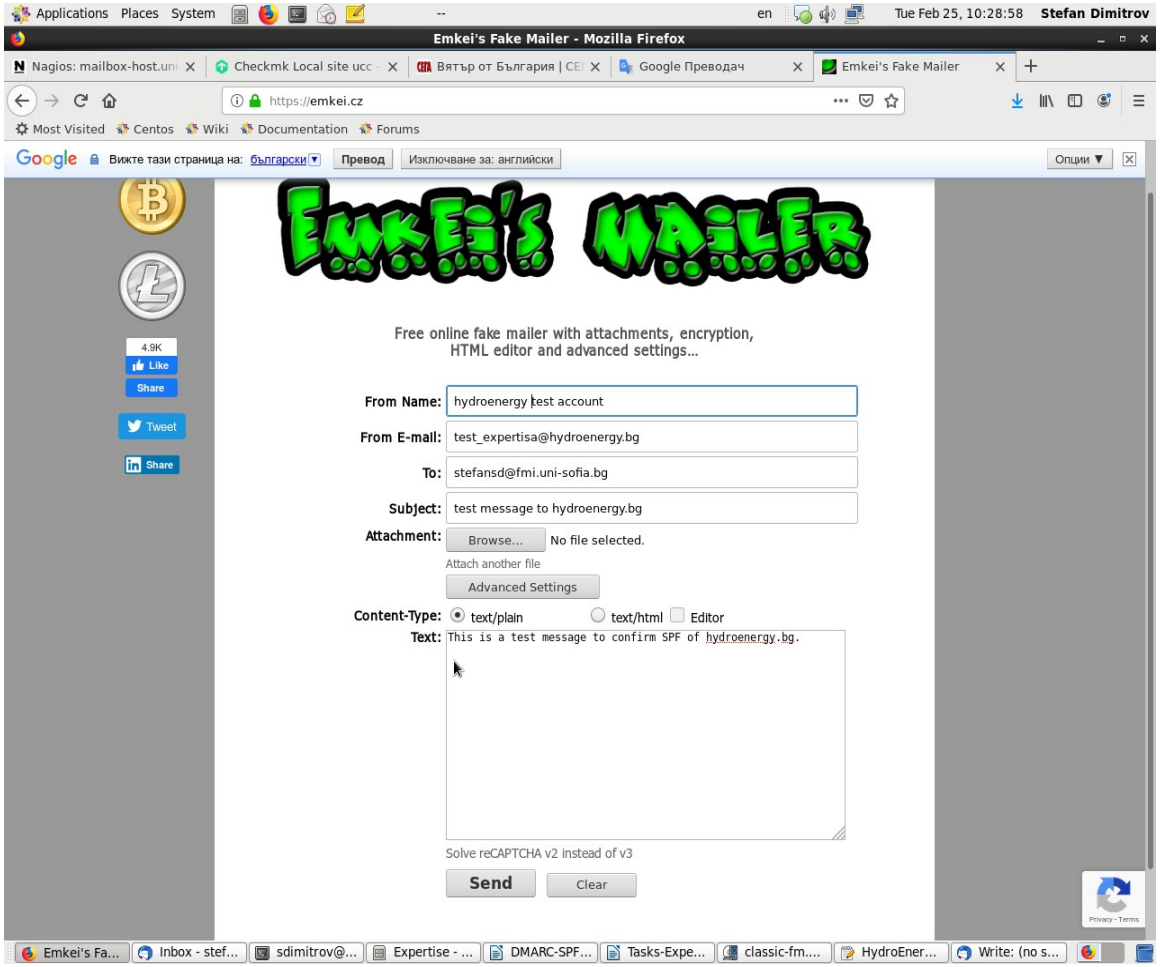
Изпращане на фалшиви електронни писма

Съществуват 10-ки (100-ци) интернет сайтове, които позволяват изпращането на електронни писма без знанието на титулярите на ел. поща. Т.нар. „**Free online fake mailer**“.

Долният линк сочи към публикация, описваща 19-те най-добри, според автора, генератори на фалшива електронна поща:

https://www.guru99.com/free-fake-email-address-generators.html

Пример: https://emkei.cz/



Решението: SPF

SPF (Sender Policy Framework):

https://www.dmarcanalyzer.com/spf/

предпазва от подправяне на адреса на подателя на e-mail и др. зловредни действия, т. нар email spoofing.

Политиката SPF се активира чрез вмъкване на специален SPF запис в зоновия файл на нашия домейн.

В този запис са описани всички авторизирани хостове по имена или IP адреси, които имат право да изпращат e-mail от името на нашия домейн.

Решението: SPF

SPF записът е текстов (**TXT**) запис. Наличието му се проверява чрез инструмента dig.

Пример за домейна fmi.uni-sofia.bg:

$ dig -t txt fmi.uni-sofia.bg

```
...
;; ANSWER SECTION:
fmi.uni-sofia.bg.     600   IN    TXT   "v=spf1 mx a
ip6:2001:67c:20d0:30::1193/128
ip6:2001:67c:20d0:30::1145/128
ip6:2001:67c:20d0:30::1147/128 ip4:62.44.101.193/32
ip4:62.44.101.114/32 ip4:62.44.101.144/32
ip4:62.44.101.145/32 ~all"
```

Решението: SPF

2001:67c:20d0:30::1193, 62.44.101.193 - petri.fmi.uni-sofia.bg.

2001:67c:20d0:30::1145 (::1147) - lists.fmi.uni-sofia.bg.

62.44.101.144 – email.fmi.uni-sofia.bg.

Пробвайте от https://emkei.cz/ да изпратите писмо от адреса си в @uni-sofia.bg до друг ваш адрес. Ще го получите ли?

Автентикация на ел. поща с DKIM

Заедно със SPF се прилага и **DKIM** (Domain Keys Identified Mail)

https://www.mailjet.com/blog/news/setting-up-dkim-step-by-step-a7d0a0ec-c4aa-4b5b-aeb5-a06361aa2e51/

На ел. поща се дава цифров подпис - DKIM подпис, който се добавя криптиран във Full Header на съобщението.

В Linux ключът се генерира със ssh-keygen и се поставя като TXT запис в зоновия файл на нашия домейн.

Пример с fmi.uni-sofia.bg:

```
61608962-2B93-11E7-9AFE-02D6DC558D50._domainkey TXT
"v=DKIM1\; k=rsa\; " "p=..."
```

CAPTCHA

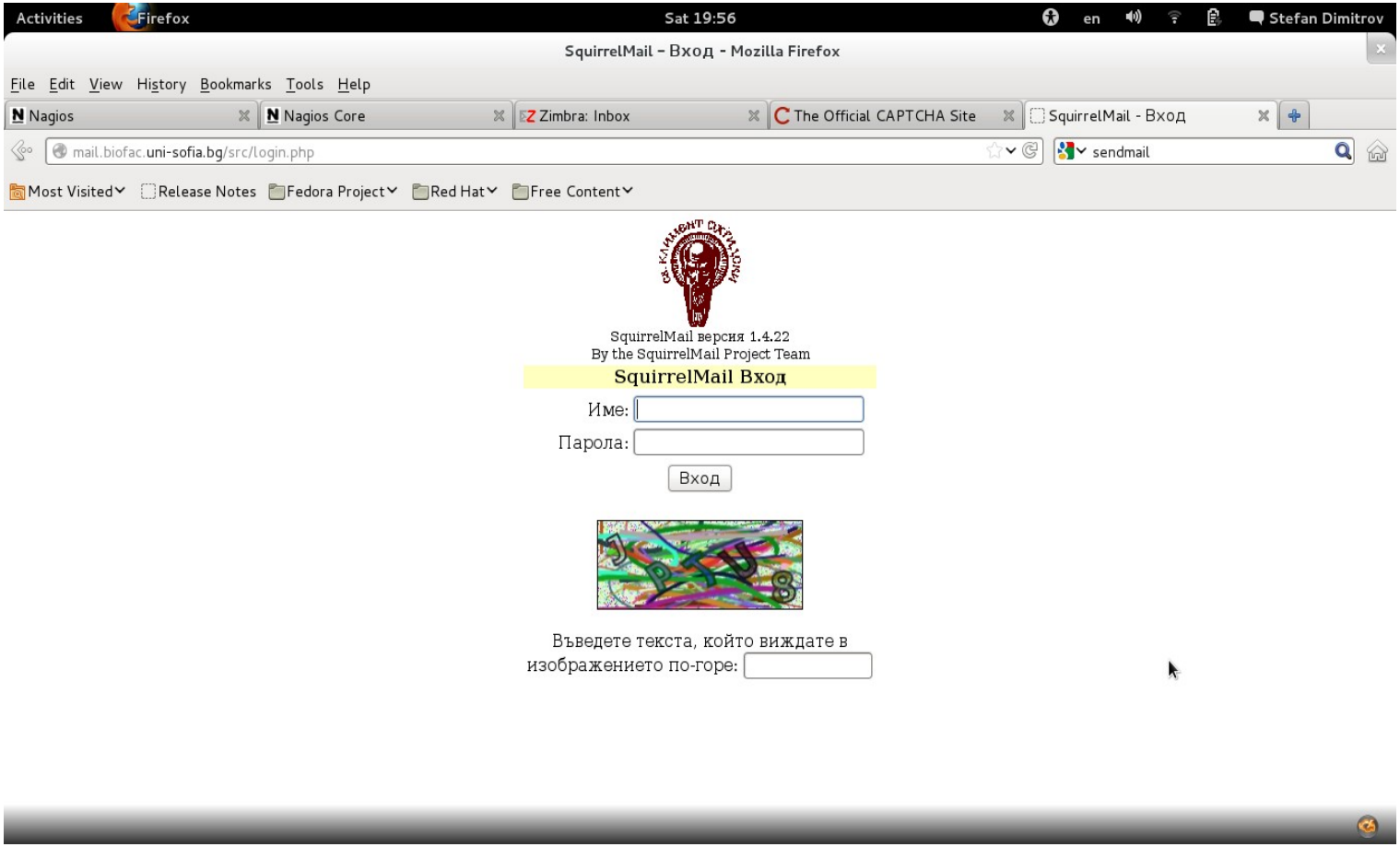
CAPTCHA

CAPTCHA (**C**ompletely **A**utomated **P**ublic **T**uring Test To Tell **C**omputers and **H**umans **A**part) е създадена през 2000 г. от Luis von Ahn, Manuel Blum, Nicholas Hopper и John Langford (Carnegie Mellon University).

Програма за защита на web сайтове (вкл. Webmail), генерираща тестове, които могат да се изпълняват от хора, но не и от компютърни програми.

Напр., разкривен текст.

Squirrelmail: captcha + lockout



Squirrelmail: captcha + lockout

cd /usr/share/squirrelmail/plugins/lockout/

cd data

vi config.php

...

логин опити на ip - max 10 за 15 min, при Превишаване – 30 min lockdown

$max_login_attempts_per_IP = '10:15:30';

...

Squirrelmail: captcha + lockout

да се активира captcha - след 5 неуспешни опита в рамките на 10мин;

captcha ще е активна за 30мин;

1=при успешно логване captcha се декативира

$activate_CAPTCHA_after_failed_attempts = '5:10:30:1';

Squirrelmail: captcha + lockout

cd /usr/share/squirrelmail/plugins/captcha/

vi config.php

има подробен списък с видовете captcha и изисванията,настройките за тях

...

```
 * @package plugins
 * @subpackage captcha
```

Обучение на SpamAssassin

За да ни е полезен SpamAssassin, трябва да обучаваме Бейсовия (Bayesian) му филтър.

Който ще сравнява предишно съдържание с известни вече **spam** и **ham** пощи, за да реши кое точно е spam.

(***ham*** *е точно обратното на spam, т.е нормалната поща.*)

Теорема на Бейс

Теоремата на Бейс е на името на Томас Бейс (**Thomas Bayes**), британски математик и презвитериански пастор.

Използва се за изчисляване на вероятността за настъпване на дадено събитие, след като вече е известна част от информацията за него.

Теорема на Бейс

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

където

P(A) – вероятност за настъпване на събитието A;

P(A|B) – Условна вероятност за настъпване на събитието A при положение, че събитието B е настъпило (апостериорна вероятност);

P(B|A) – Условна вероятност за настъпване на B при положение, че A е настъпило;

P(B) – вероятност за настъпване на събитието B.

Анти-спам филтри по теоремата на Бейс

Идеята е предложена за пръв път от английския програмист Пол Греъм.

$$\Pr(\text{spam}|\text{words}) = \frac{\Pr(\text{words}|\text{spam})\Pr(\text{spam})}{\Pr(\text{words})}$$

Pr(spam|words) - вероятността дадено съобщение да е спам, при положение че съдържа определени думи и изрази в него;

Pr(words|spam) - вероятността тези думи или изрази да се съдържат в спам-съобщение;

Pr(spam) - броят на спамовете към общия брой на съобщенията, т.е. вероятността всяко съобщение да е спам;

Pr(words) - вероятността тези думи да бъдат намерени в нормално електронно съобщение.

Как обучаваме SpamAssassin

Обучението на SpamAssassin е ефективно, само ако разполагате с достатъчен обем spam и ham.

За целта има на разположение онлайн бази от данни за първоначално "захранване" на Бейсовата база от данни на SpamAssassin.

След приключване на първоначалното обучение препоръчително е редовно да се обучава SpamAssassin, тъй като спамът постоянно се променя и "развива".

Може да се поддържа собсвена база от данни, напр. в СУ: **dnsbl.uni-sofia.bg**.

Sa-learn - нструмент за обучение на SpamAssassin

По дефолт от директория със spam и/или ham пощи добавя техните токъни към БД.

Токънът е последователност от думи или знаци, които нормално се срещат в spam или ham.

Дадените по-долу команди ще научат за spam и ham от съответните папки с имейли:

```
$ sa-learn --spam /path/to/spam/folder
$ sa-learn --ham /path/to/ham/folder
```