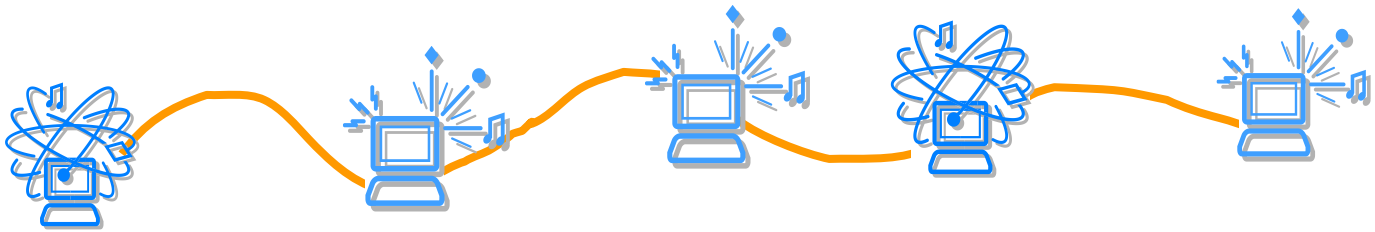
Segurança em Redes

Correio electrónico – Segurança



Redes de Comunicação de Dados

Departamento de Engenharia da Electrónica e das Telecomunicações e de Computadores

Instituto Superior de Engenharia de Lisboa

Protocolos relacionados com o email

- RFC 821 – *Simple Mail Transport Protocol* (SMTP)
- RFC 822 – *Standard for the format of ARPA Internet text messages*
- RFC 2821 - *Simple Mail Transfer Protocol*
- *RFC 2822 - Internet Message Format*
- RFC2045 - *Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies*
- RFC2046 - Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types
- RFC2047, RFC2048, RFC2049 – Outras normas referentes ao MIME

- MAIL <SP> FROM:<reverse-path> <CRLF>
  - Fornece o *reverse-path* o qual pode ser usado para reportar erros.
  - Se aceite o receptor SMTP retorna a resposta 250 OK.
  - O <reverse-path> pode conter mais do que apenas uma *mailbox*. O <reverse-path> é uma lista "*reverse source routing*" de *hosts* e *mailbox* de origem. O primeiro *host* no <reverse-path> deve ser o *host* a enviar este comando.
- RCPT <SP> TO:<forward-path> <CRLF>
  - Se aceite, o receptor SMTP retorna uma resposta 250 OK, e guarda o *forward-path*.
  - Se o receptor for desconhecido o receptor SMTP retorna uma resposta 550 Failure.
- DATA <CRLF>
  - Se aceite, o receptor SMTP retorna 354 como resposta intermédia e considera todas as linhas que se seguem como sendo a mensagem de texto..
  - Quando o fim do texto é recebido o receptor SMTP envia uma resposta 250 OK.
  - SMTP indica o fim dos dados do *mail* enviando uma linha contendo apenas um ponto. É utilizado um mecanismo para assegurar a transparência de forma a evitar a interferência com o texto do utilizador enviado no *mail*.
  - Note-se que os dados do *mail* incluem itens como *Date*, *Subject*, *To*, *Cc*, *From*
  - Se aceite pelo receptor SMTP retorna uma resposta 250 OK. O comando DATA só falha se se a transacção ficar incompleta, por exemplo falharem os destinatários ou os recursos não estarem disponíveis..

- Este exemplo SMTP mostra o *mail* enviado por Smith no host Alpha.ARPA, para Jones, Green e Brown no *host* Beta.ARPA.
- Assume-se que o Alpha contacta o host Beta directamente.
  - S: MAIL FROM:<Smith@Alpha.ARPA>
  - R: 250 OK
  - S: RCPT TO:<Jones@Beta.ARPA>
  - R: 250 OK
  - S: RCPT TO:<Green@Beta.ARPA>
  - R: 550 No such user here
  - S: RCPT TO:<Brown@Beta.ARPA>
  - R: 250 OK
  - S: DATA
  - R: 354 Start mail input; end with <CRLF>.<CRLF>
  - S: Blah blah blah...
  - S: ...etc. etc. etc.
  - S: <CRLF>.<CRLF>
  - R: 250 OK

- Este *mail* foi aceite por Jones e Brown. Green não tem uma caixa de correio no *host* Beta.

- Os dois comandos seguintes são usados na abertura e fecho do canal de transmissão:
  - HELO <SP> <domain> <CRLF>
  - QUIT <CRLF>
- No comando HELO o *host* que envia o comando identifica-se a ele próprio, o comando pode ser interpretado como "Hello, I am <domain>".
- Exemplo da abertura de uma ligação (“Connection Opening”)
  - R: 220 BBN-UNIX.ARPA Simple Mail Transfer Service Ready
  - S: HELO USC-ISIF.ARPA
  - R: 250 BBN-UNIX.ARPA

- Exemplo de fecho de uma ligação (“Connection Closing ”)
  - S: QUIT
  - R: 221 BBN-UNIX.ARPA Service closing transmission channel

Comandos SMTP

HELO <SP> <domain> <CRLF>

MAIL <SP> FROM:<reverse-path> <CRLF>

RCPT <SP> TO:<forward-path> <CRLF>

DATA <CRLF>

RSET <CRLF>

SEND <SP> FROM:<reverse-path> <CRLF>

SOML <SP> FROM:<reverse-path> <CRLF>

SAML <SP> FROM:<reverse-path> <CRLF>

VRFY <SP> <string> <CRLF>

EXPN <SP> <string> <CRLF>

HELP [<SP> <string>] <CRLF>

NOOP <CRLF> QUIT <CRLF>

TURN <CRLF>

Códigos de resposta por grupos de funções

- 500 Syntax error, command unrecognized [This may include errors such as command line too long]
- 501 Syntax error in parameters or arguments
- 502 Command not implemented
- 503 Bad sequence of commands
- 504 Command parameter not implemented
- 211 System status, or system help reply
- 214 Help message [Information on how to use the receiver or the meaning of a particular non-standard command; this reply is useful only to the human user]
- 220 <domain> Service ready
- 221 <domain> Service closing transmission channel
- 421 <domain> Service not available, closing transmission channel [This may be a reply to any command if the service knows it must shut down]
- 250 Requested mail action okay, completed
- 251 User not local; will forward to <forward-path>
- 450 Requested mail action not taken: mailbox unavailable [E.g., mailbox busy]
- 550 Requested action not taken: mailbox unavailable [E.g., mailbox not found, no access]
- 451 Requested action aborted: error in processing
- 551 User not local; please try <forward-path>
- 452 Requested action not taken: insufficient system storage
- 552 Requested mail action aborted: exceeded storage allocation
- 553 Requested action not taken: mailbox name not allowed [E.g., mailbox syntax incorrect]
- 354 Start mail input; end with <CRLF>.<CRLF>
- 554 Transaction failed

SMTP

- **Estabelecimento da ligação**

  O canal de transmissão SMTP é uma ligação TCP estabelecida entre o processo que envia no porto U e o processo que recebe no porto L. Esta ligação *full-duplex* é utilizada como canal de transmissão. Este protocolo tem atribuído o porto de serviço 25, isto é L=25.

- **Transferência de dados**

  A ligação TCP suporta a transmissão de bytes (8 bits). Os dados SMTP são caracteres ASCII a 7 bits. Cada carácter é transmitido como um byte de 8 bits com o bit de maior peso igual a zero.

Ameaças à segurança dos *email*

- Podem-se distinguir dois tipos de ameaças à segurança do *email*:
  - Ameaças à segurança do próprio *email*
  - Ameaças a uma organização tornadas possíveis devido ao uso do *email*
- São possíveis outras classificações!
- Lista de ameaças não exaustiva!

Ameaças ao *email*

- **Perda de confidencialidade**
  - Os *emails* são enviados em claro através de redes públicas.
  - Os *emails* são guardados em clientes e servidores potencialmente inseguros.
  - Assegurar a confidencialidade pode ser importante para os *email* trocados dentro de uma organização.

- **Perda de integridade**
  - Sem protecção da integridade nos *emails*; o corpo da mensagem pode ser alterada em trânsito ou no servidor de *email*.

Ameaças ao *email*

- **Falta de autenticação da origem dos dados**
  - Este *email* vem mesmo da pessoa que consta no campo **From:** ?
  - Quantos manuel.silva existem por aí?
  - Aceder ao servidor SMTP directamente por telnet permite falsificar todos os campos do *email*
  - O *email* também pode ser alterado em trânsito
  - Mesmo que o campo **From:** pareça bem, quem é que ligou fazendo-se passar por manuel.silva quando o *email* foi escrito?
  - A partilha de *passwords* de *email* ainda acontece.

Ameaças ao *email*

- **Falta de não-repudiação**
  - Posso confiar e agir de acordo com o conteúdo? (integridade)
  - Se sim, pode quem enviou negar ter enviado o *email*? Quem é responsável se eu agir de acordo com o *email*?
  - Exemplo de uma transação de acções utilizando *email.*

- **Falta de notificação de recepção (recibo)**
  - O destinatário do meu *email* recebeu o dito e actuou de acordo?
  - Uma mensagem marcada localmente como ‘sent’ pode não ter sido recebida

Ameaças permitidas pelo *email*

- **Divulgação de informação sensível**
  - É mais fácil distribuir informação via *email* do que em papel via *snail mail.*
  - A divulgação pode ser deliberada (e maliciosa) ou sem intenção.
  - A divulgação pode ser interna ou externa à empresa.
  - A divulgação pode ser de informação pessoal, inapropriada, sensível, comercial ou proprietária.
  - Pode levar à perda de reputação, ao despedimento de funcionários ou mesmo a processos no tribunal.

Ameaças permitidas pelo *email*

- **Exposição de sistemas a código maldoso**
  - Hoje, o *email* é o vector principal de entrada de vírus nos computadores
  - O código que se auto replica incluido nos *emails* explora as facilidades/vulnerabilidades do cliente do *email*
    - *Scripts* Visual Basic
    - Javascript em *emails* formatados em HTML
    - Anexos .exe de programas desconhecidos
  - Quase sempre (mas não sempre) requer a interacção com o utilizador para propagar um vírus de *email*
  - A dispersão do vírus pode resultar num DoS (*Denial of Service*)

Ameaças permitidas pelo *email*

- **Exposição de sistemas a ataques DoS**
  - Servidor de *email* ligado à rede mais exposta ao exterior pode estar mais vulnerável a ataques DoS
  - Mais relevante com o crescimento da dependência no *email* como ferramenta de comunicação
  - Por exemplo, um *worm* virulento usando uma percentagem elevada da capacidade da rede para se expandir pode reduzir a capacidade de utilizar eficientemente o *email*, bem como reduzir a velocidade de acesso para aceder a páginas *web*

Ameaças permitidas pelo *email*

- **Exposição de individuos a ataques DoS**
  - "*Mail bombing*" e *spam* excessivo
  - Individuos ficam tão submersos pelos *emails* que chegam a deixar de os ler e passam a outros métodos de comunicação.

Ameaças permitidas pelo *email*

- ***Spamming***
  - O *spam* gasta largura de banda e faz decrescer a produtividade
  - O *hotmail* e outros sistemas de *email* gratuitos são vítimas preferenciais dos *spammers*
  - Mais de 50% dos *emails* são *spam,* há opiniões de que é 80 a 90%
  - Existe legislação *anti-spam* a ser criada em muito países
    - Federal CAN-SPAM act em vigor nos EUA desde 1/1/2004.
    - Não torna o *spamming* fora da lei, mas controla o seu uso.
    - Primeira condenação em Setembro de 2004, Nicholas Tombros.
      - *Spamming* + *war-driving*.
    - Ver http://www.spamlaws.com/ para detalhes da leis.
    - A efectividade do CAN-SPAM e legislação semelhante ainda em questão

Ameaças permitidas pelo *email*

- ***Relaying* e listas negras (*blacklisting*)**
  - Má configuração das capacidades de *relaying* permite que servidores de *email* sejam utilizados para explorar o *spamming*
  - O servidor culpado pode acabar por ser colocado na lista negra ("***Open Relay Blacklist***")
  - Resulta em que todos os *emails* enviados desse servidor serão bloqueados pelos servidores que utilizarem a lista negra.

Ameaças permitidas pelo *email*

- **Acesso não autorizado a sistemas**
  - Os servidores de *email* (SO e aplicações) podem ter muitas vulnerabilidades de segurança; eles estão ligados a redes mais vulneráveis
  - Alvos perfeitos para os *hackers*
  - Leva a que os servidores de *email* sirvam de plataforma de ataque a outros sistemas (aos da própria empresa ou de outras)
  - Perda consequente de reputação e processos legais por perdas e danos

Ameaças permitidas pelo *email*

- Mais ameaças?

Normas e produtos para melhorar a segurança do email

- **S/MIME, PGP, SPF e *DomainKeys***
  - As duas primeiras destinam-se à segurança entre utilizadores finais de maneira a se poder garantir autenticação, integridade e confidencialidade extremo-a-extremo.
  - As duas últimas pretendem ser mais uma medida de autenticação e integridade, mas apenas entre servidores de *email*.
- Há outras normas com os "pés para a cova", também ditas defuntas: PEM (*privacy enhanced mail*), X.400.
  - Partes destas persistem: O PEM introduziu a codificação base64, o X.400 levou às normas dos certificados digitais X.509
- Muitos produtos comerciais:
  - Hushmail (www.hushmail.com),
  - XenoMail,
  - *Identity-based secure email* (www.voltagesecurity.com), …

Normas e produtos de para melhorar a segurança do *email*

- S/MIME (*Secure Multi-Purpose Internet Mail Extensions*):
  - É um método seguro de enviar correio electrónico que usa o protocolo RSA incluído nas últimas versões dos *browsers* da Microsoft e da Netscape assim como em outros produtos de correio electrónico
  - Consta de cinco partes:
    - RFC 3369, *Cryptographic Message Syntax*
    - RFC 3370, *Cryptographic Message Syntax (CMS) Algorithms*
    - RFC 2633, *S/MIME Version 3 Message Specification*
    - RFC 2632, *S/MIME Version 3 Certificate Handling*
    - RFC 2631, *Diffie-Hellman Key Agreement Method*

    e um protocolo adicional: RFC 2634, *Enhanced Security Services for S/MIME*
- • PGP/MIME
  - Semelhante ao S/MIME mas baseado em PGP e construído com base nos RFCs:
    - RFC 1991, *PGP Message Exchange Formats*
    - RFC 2015, *MIME Security with Pretty Good Privacy*

Segurança no SMTP

- A forma de envio de mensagens por SMTP (RFC 2821) não contempla segurança.
  - Permite falsificação do remetente
  - O conteúdo das mensagens circula em claro
  - Os servidores estão vulneráveis a serem usados como "amplificadores" e/ou repetidores de mensagens de outrem (*open relay*)
  - Dá a possibilidade de, por exemplo, através de acesso via Telnet se poderem fazer muitas falcatruas.

Validação de remetentes no SMTP (extensões)

- Minimiza possibilidades de falsificação de mensagens
  - Não consegue eliminar esta possibilidade por a falsificação poder ainda ocorrer noutro **servidor "promíscuo".**

- Formas definidas de autenticação (RFC 2554 - *SMTP Service Extension for Authentication*):
  - **PLAIN** – Em claro (utilizador e palavras-chave enviados numa única linha codificados em base64)
  - **LOGIN** – Em claro (utilizador e palavras chave codificadas em base64)
  - **CRAM-MD5** – Baseada em *challenge-response, hash* seguro

- Pode-se verificar quais os tipos de autenticação suportados por um servidor fazendo *telnet* para o endereço do servidor de *email*, porto 25, e enviando o comando EHLO.

Tipos de autenticação

- Antes de um cliente tentar se autenticar para uma entrega SMTP envia primeiro um comando SMTP EHLO. O servidor SMTP responde listando as extensões ESMTP que são suportadas por esse servidor. Atualmente são utilizadas três tipos de extensões de autenticação (ESMTP AUTH) que são designadas por PLAIN, LOGIN e CRAM-MD5.

AUTH PLAIN

O mecanismo de autenticação PLAIN especifica que sejam enviadas três *strings* com o comando AUTH. A segunda e a terceira são respectivamente o *login* e a *password*. Se o teste destes tiver sucesso o servidor prossegue.

AUTH LOGIN

Este método de autenticação envolve uma conversação entre servidor de *email* e cliente na qual o servidor de *email* apresenta ao cliente dois *prompts*, usualmente Username: e Password: . Estes *prompts* são codificado em base64 tal como as respostas aos mesmos, embora o método AUTH LOGIN não esteja documentado no RFC 2554 é um método muito utilizado.

AUTH CRAM-MD5

O método de autenticação CRAM-MD5 é o mais seguro dos três tipos de autenticação aqui descritos. O servidor envia ao cliente uma *string* de desafio codificada em CRAM-MD5 e o cliente responde usando a *string* de desafio para construir a resposta que consiste no nome do utilizador e no *digest* CRAM-MD5 da *string* de desafio e da *password*. O servidor calcula a resposta CRAM-MD5 que deveria receber e compara com a recebida do cliente. Se a comparação das duas tiver sucesso a autenticação tem sucesso.

Autenticação SMTP/PLAIN

220 mail.example.com ESMTP Postfix
EHLO anotherhost.com
250-mail.example.com
250-PIPELINING
250-SIZE 10240000
250-VRFY
250-ETRN
250-STARTTLS
250-AUTH PLAIN LOGIN GSSAPI DIGEST-MD5 CRAM-MD5
250-XVERP
250 8BITMIME
AUTH PLAIN dXNlcm5hbWUAdXNlcm5hbWUAcGFzc3dvcmQ=
[username\0username\0password]
235 Authentication successful
QUIT
221 Bye

Legenda:

Decodificação do texto anterior em base64
Respostas do servidor
Comandos do cliente (envelope)

Autenticação SMTP/LOGIN

```
220 IPLNetSMTPi ESMTP
EHLO pedro
250-IPLNetSMTPi
250-PIPELINING
250-8BITMIME
250-SIZE 10485760
250-AUTH LOGIN CRAM-MD5 PLAIN
250-AUTH=LOGIN CRAM-MD5 PLAIN
250 OK
AUTH LOGIN
334 VXNlcm5hbWU6 [Username:]
cHJpYmVpcm9AZGVldGMuaXNlbC5pcGwucHQ
      = [pribeiro@deetc.isel.ipl.pt]
334 UGFzc3dvcmQ6 [Password:]
QmVtRXNjb25kaWRh [BemEscondida]
235 ok, go ahead (#2.0.0)
MAIL FROM: <pribeiro@deetc.isel.ipl.pt>
250 ok
RCPT TO: <pribeiro@isel.ipl.pt>
250 ok
DATA
354 go ahead
From: "Pedro Ribeiro"
      <pribeiro@deetc.isel.ipl.pt>
To: <pribeiro@isel.ipl.pt>
Subject: Teste2
Date: Tue, 19 Oct 2004 23:14:22 +0100
MIME-Version: 1.0
Content-Type: text/plain;
.charset="us-ascii"
Content-Transfer-Encoding: 7bit
Thread-Index:
      AcS2KPxbXX3coTxyRRKbp0QfAC0Qdw==


.
250 ok 1098224064 qp 14425
QUIT
221 IPLNetSMTPi Closing
```

Legenda:

- Respostas do servidor
- Comandos do cliente (envelope)
- Corpo da mensagem
- Decodificação de mensagens em base64

Autenticação SMTP/CRAM-MD5

```
220 IPLNetSMTPi ESMTP
EHLO 99.29.79.10.in-addr.arpa
250-IPLNetSMTPi
250-PIPELINING
250-8BITMIME
250-SIZE 10485760
250-AUTH LOGIN CRAM-MD5 PLAIN
250-AUTH=LOGIN CRAM-MD5 PLAIN
250 OK
AUTH CRAM-MD5
334
    PDE5NTM5LjEwOTgyMjUyMjVAc210cC1vdX
    QubmV0LmlwbC5wdD4=
    [<19539.1098225225@smtp-out.net.ipl.pt>]
cHJpYmVpcm9AbmV0LmlwbC5wdCAxNzliOWEzN
    jk4YTk1ODdiZjFlMGMzYmMwZTU1NWU1NW
    == [pribeiro@net.ipl.pt
    179b9a3698a9587bf1e0c3bc0e555e55]
235 ok, go ahead (#2.0.0)
MAIL FROM:<pribeiro@net.ipl.pt> SIZE=416
250 ok
RCPT TO:<pribeiro@deetc.isel.ipl.pt>
250 ok
DATA
354 go ahead
```

```
Date: Tue, 19 Oct 2004 23:33:44 +0100
From: Pedro Ribeiro <pribeiro@net.ipl.pt>
Reply-To: Pedro Ribeiro <pribeiro-
    reply@net.ipl.pt>
To: pribeiro@deetc.isel.ipl.pt
Subject: Teste CRAM-MD5
MIME-Version: 1.0
Content-Type: text/plain; charset=ISO-8859-15
Content-Transfer-Encoding: 8bit


.
250 ok 1098225226 qp 19541
RSET
250 flushed
QUIT
221 IPLNetSMTPi Closing
```

Legenda:

- Respostas do servidor
- Comandos do cliente (envelope)
- Corpo da mensagem
- Decodificação de mensagens em base64

SMTP sobre túnel TLS (cifra)

- Com porto dedicado (465/TCP)
  - É estabelecido um túnel TLS (SSL) sobre o qual corre o SMTP
  - As autenticações PLAIN e LOGIN passam a estar protegidas

- Com activação de segurança sobre SMTP
  - Usa o comando STARTTLS (RFC3207)

SMTP - Túnel TLS (STARTTLS)

220 smtp.example.org ESMTP Sendmail 8.13.0/8.13.0
EHLO client.example.org
250-smtp.example.org Hello client.example.org [192.0.2.7], pleased to meet you
250-ENHANCEDSTATUSCODES
250-PIPELINING
250-8BITMIME
250-SIZE 16180340
250-DSN
250-STARTTLS
250-DELIVERBY
250 HELP
STARTTLS
220 2.0.0 Ready to start TLS
**** Conteúdo cifrado ****

Legenda:

Respostas do servidor
Comandos do cliente (envelope)

Segurança no POP3 original

- A forma de recepção de mensagens por POP3 (RFC1725- *Post Office Protocol - Version 3*) não contempla segurança.
  - O cliente autentica-se passando a identificação e palavra-chave em claro
  - O conteúdo das mensagens circula em claro

Validação de remetentes no POP3 (extensões)

- Minimiza possibilidades de roubo de identidade
  - As palavras-chave deixam de circular directamente no canal.

- Formas definidas de autenticação:
  - APOP – Baseada em *challenge-response* (utilizador e *digest* enviados numa única linha)
    - Possibilidade definida no RFC 1734 e incluída na revisão do protocolo do RFC 1939
      - **APOP name digest** (digest: MD5<process-ID.clock@hostname>,secret)
  - CRAM-MD5 – Baseada em *challenge-response,* idêntica à usada no SMTP.
    - Mais segura que o APOP
    - Definida no RFC 1734 e RFC 2095

POP3 sobre túnel TLS (cifra)

- Com porto dedicado (995/TCP)
  - É estabelecido um túnel TLS (SSL) sobre o qual corre o POP3
  - A autenticação tradicional em claro passa a estar protegida pelo TLS
- Com activação de segurança sobre POP3
  - Usa o comando STARTTLS (RFC2595)

Autenticação POP3 original

+OK <3262.1098353259@pop.net.ipl.pt>
USER pribeiro@net.ipl.pt
+OK
PASS MuitoSecreta
+OK
STAT
+OK 0 0
QUIT
+OK

Legenda

- Respostas do servidor
- Comandos do cliente

Autenticação POP3/APOP

+OK <45012.1098353411@pop.net.ipl.pt>
APOP pribeiro@net.ipl.pt 52ba727eb1a49878d7a08d221b3e7094
+OK
STAT
+OK 0 0
QUIT
+OK

Legenda:

- Respostas do servidor
- Comandos do cliente

Autenticação POP3/CRAM-MD5

+OK <41432.1098352762@pop.net.ipl.pt>
AUTH CRAM-MD5
+ PDMxMjUwLjEwOTgzNTI3NjJAcG9wLm5ldC5pcGwucHQ+
cHJpYmVpcm9AaXNlbC5pcGwucHQgOGMzMGQwYmFiOTdlZWEyNzVlOTc2MTU0
MDEyOGZiMGE= [pribeiro@isel.ipl.pt 8c30d0bab97eea275e9761540128fb0a]
+OK
STAT
+OK 0 0
QUIT
+OK

Legenda:

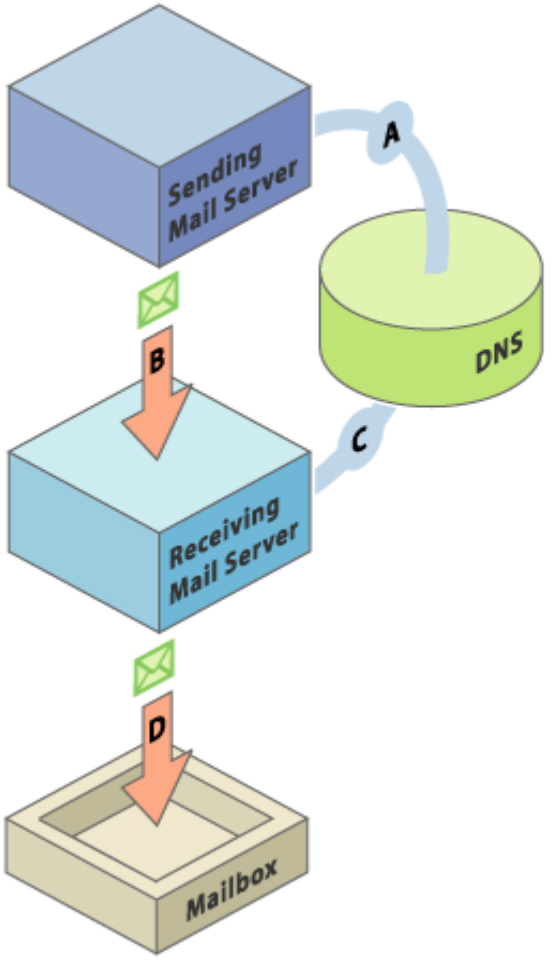
- Respostas do servidor
- Comandos do cliente (envelope)
- Decodificação de mensagens em base64

Medidas anti-SPAM - SPF

- **SPF - *Sender Policy Framework*** (*draft*)
  - Os administradores dos servidores de SMTP/DNS publicam no DNS informação acerca de quais são os servidores válidos para envio de *email* de cada domínio que gerem
  - Os servidores de email ao receberem uma mensagem verificam se o servidor que está a tentar o envio está autorizado a enviar *email* com o domínio do remetente por consulta ao DNS (temporariamente usou um "*Record*" TXT até lhe ser reservado um específico. Ex: alunos.isel.ipl.pt. 3600 IN TXT "v=spf1 ip4:193.137.220.0/25 ip4:62.48.232.168 -all"
  - Atualmente existe no DNS o registo SPF, tipo 99, mas existe alguma polémica pelo que o TXT continua a ser usado
  - Referência: http://spf.pobox.com/draft-ietf-marid-protocol-00.txt

DomainKeys: Anti-SPAM e anti-falsificação de endereços (*anti-spoofing*)

- A chave pública é publicado no servidor de DNS (A).
- As mensagens são assinadas com chave privada pelo servidor de *email* que envia as mensagens (B).
- Ao chegar as mensagens são verificadas para se saber se o seu conteúdo foi alterado. Sendo a chave pública do emissor obtida do servidor de DNS (C). São verificados também os campos From: e Sender: para se verificar se o *email* veio mesmo donde diz que veio.
- Se a mensagem não tiver sido alterada e cumprir as outras regras associadas à politica de segurança dos *emails* é enviada para o destinatário (D).
- O DomainKeys é mais seguro do que o SPF pois utiliza assinatura digital.



http://antispam.yahoo.com/domainkeys

DKIM

- Evolução do DomainKeys com mistura do SPF

Medidas anti-SPAM – SenderID

- Desenvolvido pela Microsoft
  - A comunidade "*Open Source*" tem colocado reticências por existirem aspectos sujeitos a patentes
  - É uma versão melhorada do SPF com validações adicionais incluídas em cabeçalhos