FREDRICK OCHIENG
DEPARTMENT OF ICT
ZETECH UNIVERSITY

Email Security

MR FREDRICK OCHIENG
0700700763

Contents

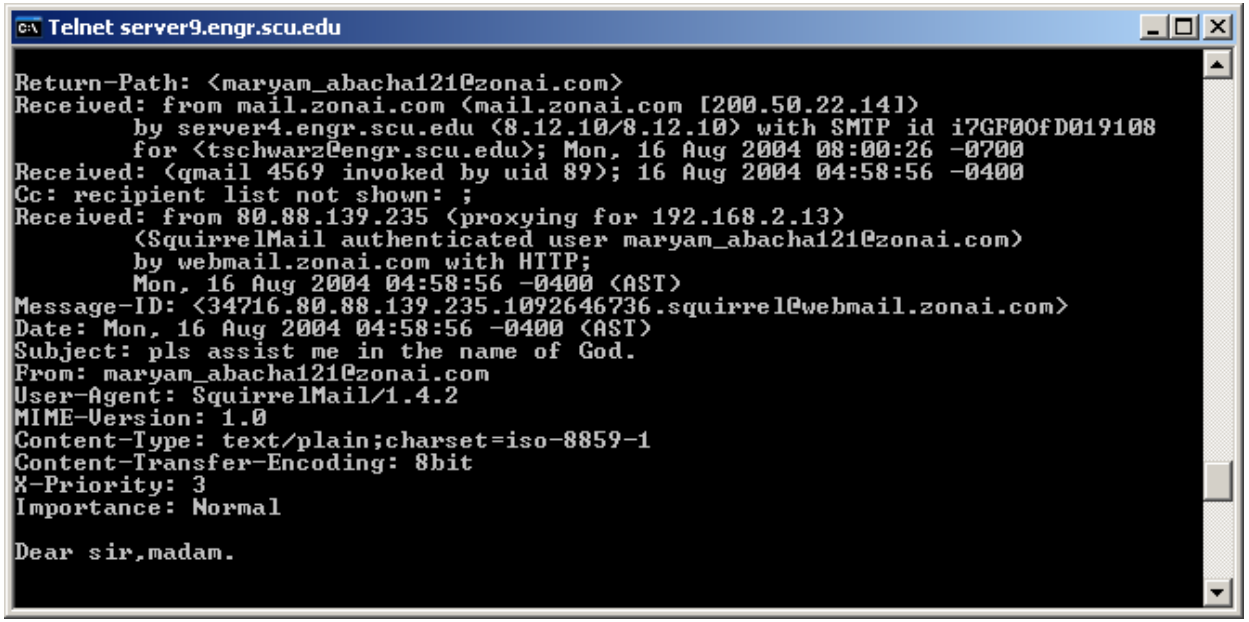
- Why?
  - How to forge email?
  - How to spot spoofed email.
- Distribution Lists
  - The twist that makes email authentication … interesting.
- Mail Infrastructure
- Security Characteristics
  - Authentication
  - Confidentiality
  - Non-repudiation
- Solutions:
  - PEM
  - S/MIME
  - PGP

E-mail Security Desires

- Current email
  - Can be easily forged.
  - Can be generated almost free of cost and used for spamming.
  - Contains no guarantee for delivery.
  - Has currently no inbuilt authentication method.

Email Fundamentals

- Email travels from originating computer to the receiving computer through email servers.
- All email servers add to the header.
- Use important internet services to interpret and verify data in a header.



Email Fundamentals

- Typical path of an email message:

Client → Mail Server → Mail Server → Mail Server → Client

MR FREDRICK OCHIENG
0700700763

Email Fundamentals: Important Services

- Verification of IP addresses:
  - Regional Internet Registry
    - APNIC (Asia Pacific Network Information Centre).
    - ARIN (American Registry of Internet Numbers).
    - LACNIC Latin American and Caribbean IP address Regional Registry.
    - RIPE NCC (Réseau IP Européens Network Coordination Centre).
  - Whois
  - www.samspade.org ← My Favorite.
  - Numerous other websites.
- Remember: Whole emails can be forged.

Email Fundamentals: Important Services

- Domain Name System (DNS) translates between domain names and IP address.
  - Name to address lookup:
    1. Parses **HOSTS** file.
    2. Asks **local nameserver**
    3. Local nameserver contacts **nameserver responsible for domain.**
    4. If necessary, contact **root nameserver.**
    5. Remote nameserver sends data back to local nameserver.
    6. Local nameserver caches info and informs client.
  - HOSTS files can be altered.
    - You can use this as a low-tech tool to block pop-ups.
  - Local nameservers can/could be tricked into accepting unsolicited data to be cached.
    - "Hilary for Senate" – case.

Email Fundamentals: Important Services

- Many organizations use Network Address Translation.
  - NAT boxes have a single visible IP.
  - Incoming I-packet analyzed according to address and port number.
  - Forwarded to interior network with an **internal** IP address.
  - Typically in the private use area:
    - 10.0.0.0 – 10.255.255.255
    - 172.16.0.0 – 172.31.255.255
    - 192.168.0.0-192.168.255.255
  - Private use addresses are never used externally.

Email Protocols:

- Email program such as outlook is a **client application.**
- Needs to interact with an email server:
  - Post Office Protocol (POP)
  - Internet Message Access Protocol (IMAP)
  - Microsoft's Mail API (MAPI)

Email Protocols:

- A mail server stores incoming mail and distributes it to the appropriate mail box.
- Behavior afterwards depends on type of protocol.
- Accordingly, investigation needs to be done at server or at the workstation.

Email Protocols:

Post Office Service	Protocol	Characteristics
Stores only incoming messages.	POP	Investigation must be at the workstation.
Stores all messages	IMAP MS' MAPI Lotus Notes	Copies of incoming and outgoing messages might be stored on the workstation or on the server or on both.
Web-based send and receive.	HTTP	Incoming and outgoing messages are stored on the server, but there might be archived or copied messages on the workstation. Easy to spoof identity.

Email Protocols: SMTP

- Neither IMAP or POP are involved relaying messages between servers.
- Simple Mail Transfer Protocol: SMTP
  - Easy, but can be spoofed easily.

Email Protocols: SMTP

How to spoof email:

telnet server8.engr.scu.edu 25
220 server8.engr.scu.edu ESMTP Sendmail 8.12.10/8.12.10; Tue, 23 Dec 2003 16:32:07 -0800 (PST)
helo 129.210.16.8
250 server8.engr.scu.edu Hello dhcp-19-198.engr.scu.edu [129.210.19.198], pleased to meet you
mail from: jholliday@engr.scu.edu
250 2.1.0 jholliday@engr.scu.edu... Sender ok
rcpt to: tschwarz
250 2.1.5 tschwarz... Recipient ok
data
354 Enter mail, end with "." on a line by itself
This is a spoofed message.
.
250 2.0.0 hBO0W76P002752 Message accepted for delivery
quit
221 2.0.0 server8.engr.scu.edu closing connection

Email Protocols: SMTP

From jholliday@engr.scu.edu Tue Dec 23 16:44:55 2003
Return-Path: <jholliday@engr.scu.edu>
Received: from server8.engr.scu.edu (root@server8.engr.scu.edu [129.210.16.8])
by server4.engr.scu.edu (8.12.10/8.12.10) with ESMTP id hBO0itpv008140
for <tschwarz@engr.scu.edu>; Tue, 23 Dec 2003 16:44:55 -0800
From: JoAnne Holliday <jholliday@engr.scu.edu>
Received: from 129.210.16.8 (dhcp-19-198.engr.scu.edu [129.210.19.198])
by server8.engr.scu.edu (8.12.10/8.12.10) with SMTP id hBO0W76P002752
for tschwarz; Tue, 23 Dec 2003 16:41:55 -0800 (PST)
Date: Tue, 23 Dec 2003 16:32:07 -0800 (PST)
Message-Id: <200312240041.hBO0W76P002752@server8.engr.scu.edu>
X-Spam-Checker-Version: SpamAssassin 2.60-rc3 (1.202-2003-08-29-exp) on

none autolearn=ham

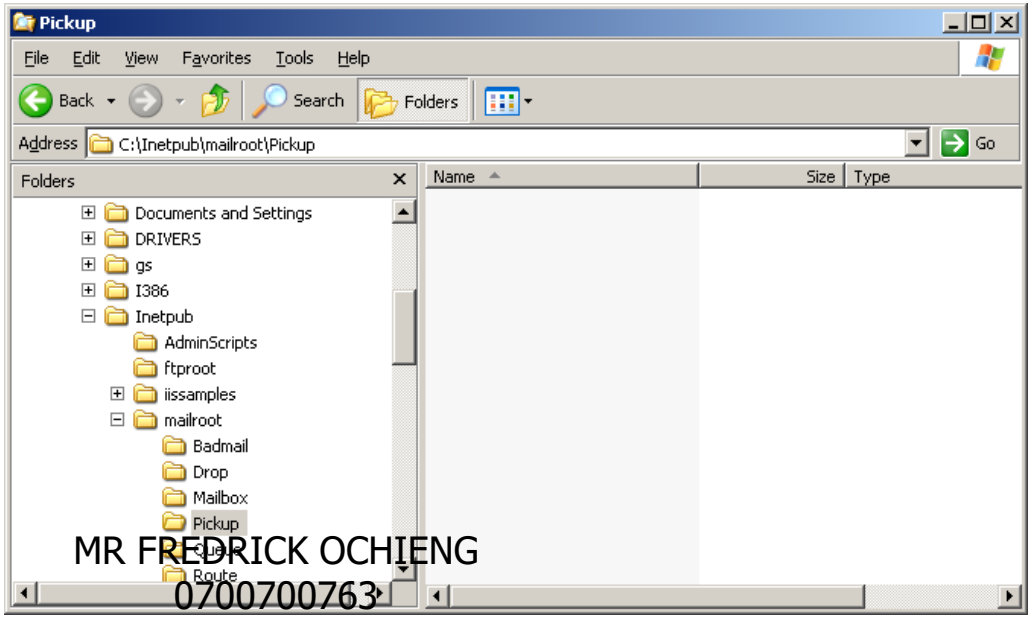
This looks very convincing.

Only hint: received line gives the name of my machine, defaulting to dhcp-19-198.

The DHCP server logs might tell you what machine this is, given the time. But you need to know the clock drift at the various machines.

Email Protocols: SMTP

- Things are even easier with Windows XP.
  - Turn on the SMTP service that each WinXP machine runs.
  - Create a file that follows SMTP protocol.
  - Place the file in Inetpub/mailroot/Pickup



Email Protocols: SMTP

To: tschwarz@
From: HolyFa

This is a spoo

From HolyFather@vatican.va Tue Dec 23 17:25:50 2003
Return-Path: <HolyFather@vatican.va>
Received: from Xavier (dhcp-19-226.engr.scu.edu [129.210.19.226])
by server4.engr.scu.edu (8.12.10/8.12.10) with ESMTP id hBO1Plpv027244
for <tschwarz@engr.scu.edu>; Tue, 23 Dec 2003 17:25:50 -0800
Received: from mail pickup service by Xavier with Microsoft SMTPSVC;
Tue, 23 Dec 2003 17:25:33 -0800
To: tschwarz@engr.scu.edu
From: HolyFather@vatican.va
Message-ID: <XAVIERZRTHEQXHcJcKJ00000001@Xavier>
X-OriginalArrivalTime: 24 Dec 2003 01:25:33.0942 (UTC) FILETIME=[D3B56160:01C3C9
BC]
Date: 23 Dec 2003 17:25:33 -0800
X-Spam-Checker-Version: SpamAssassin 2.60-rc3 (1.202-2003-08-29-exp) on
server4.engr.scu.edu
X-Spam-Level:
X-Spam-Status: No, hits=0.3 required=5.0 tests=NO_REAL_NAME autolearn=no
version=2.60-rc3

This is a spoofed message.

MR FREDRICK OCHIENG
0700700763

Email Protocols: SMTP

- SMTP Headers:
  - Each mail-server adds to headers.
  - Additions are being made at the top of the list.
    - Therefore, read the header from the bottom.
- To read headers, you usually have to enable them.

SMTP Headers

To enable headers:

- Eudora:
  - Use the Blah Blah Blah button
- Hotmail:
  - Options → Preferences → Message Headers.
- Juno:
  - Options → Show Headers
- MS Outlook:
  - Select message and go to options.
- Yahoo!:
  - Mail Options → General Preferences → Show all headers.

SMTP Headers

- Headers consists of *header fields*
  - Originator fields
    - from, sender, reply-to
  - Destination address fields
    - To, cc, bcc
  - Identification Fields
    - Message-ID-field is optional, but extremely important for tracing emails through email server logs.
  - Informational Fields
    - Subject, comments, keywords
  - Resent Fields
    - Resent fields are strictly speaking optional, but luckily, most servers add them.
    - Resent-date, resent-from, resent-sender, resent-to, resent-cc, resent-bcc, resent-msg-id

SMTP Headers

- Trace Fields
  - Core of email tracing.
  - Regulated in RFC2821.
  - When a SMTP server receives a message for delivery or forwarding, it MUST insert trace information at the beginning of the header.

SMTP Headers

- The FROM field, which must be supplied in an SMTP environment, should contain both (1) the name of the source host as presented in the EHLO command and (2) an address literal containing the IP address of the source, determined from the TCP connection.
- The ID field may contain an "@" as suggested in RFC 822, but this is not required.
- The FOR field MAY contain a list of <path> entries when multiple RCPT commands have been given.
- A server making a final delivery inserts a return-path line.

SMTP Header

- Spotting spoofed messages
  - Contents usually gives a hint.
  - Each SMTP server application adds a different set of headers or structures them in a different way.
    - A good investigator knows these formats.
  - Use internet services in order to verify header data.
    - However, some companies can outsource email or use internal IP addresses.
  - Look for breaks / discrepancies in the “Received” lines.

Server Logs

- E-mail logs usually identify email messages by:
  - Account received
  - IP address from which they were sent.
  - Time and date (beware of clock drift)
  - IP addresses

Server Logs

- Many servers keep copies of emails.
- Most servers purge logs.
  - Law-enforcement:
    - Vast majority of companies are very cooperative.
    - Don’t wait for the subpoena, instead give system administrator a heads-up of a coming subpoena.
  - Company:
    - Local sys-ad needs early warning.
    - Getting logs at other places can be dicey.

Unix Sendmail

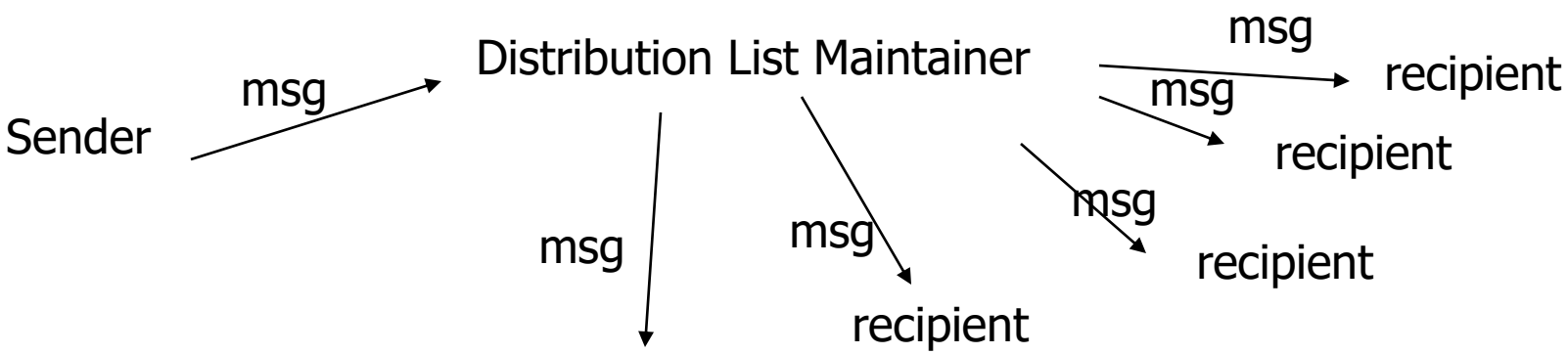
- Configuration file /etc/sendmail.cf and /etc/syslog.conf
  - Gives location of various logs and their rules.
- maillog (often at /var/log/maillog)
  - Logs SMTP communications
  - Logs POP3 events
- You can always use: locate *.log to find log files.

Conclusions up till now

- It is very easy to spoof email.
- It is possible, but hard to trace email.
  - But if the forgery happens too close to the receiver, then it is impossible.
- Basic Rule of Tracing E-mail:
  - Once the message leaves the forger's domain, SMTP headers will be accurate.

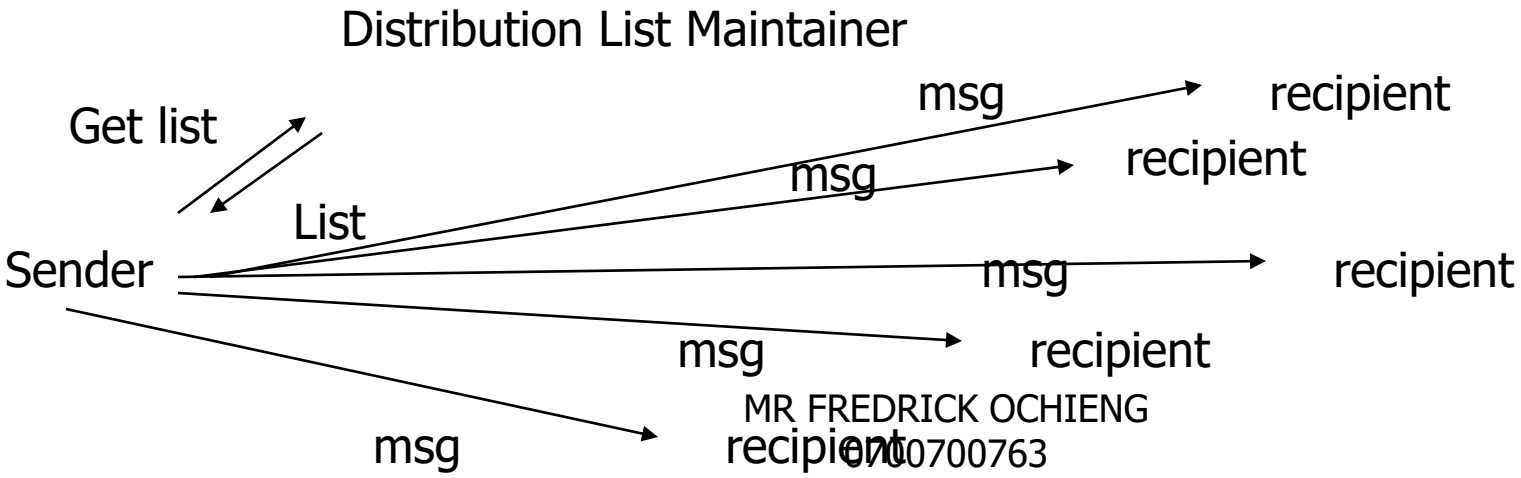
Email: Distribution List

- Simplest:
  - Single recipient per email message.
- Distribution List
  - Send mail to a **set** of recipients.
  - Remote Exploder Model



Email: Distribution List

- Distribution List
  - Send mail to a **set** of recipients.
  - Remote Exploder Model
  - Local Exploder Model



Email: Distribution List

- Local Exploder
  - Easier to prevent mail forwarding loops.
    - Caused by distribution lists contained in distribution lists.
  - Easier to prevent multiple copies of the same message.
    - By weeding out duplicates in the list.
  - Bandwidth consumption is known to user.
    - Important when we start billing per email message.

Email: Distribution List

- Remote Exploder
  - Allows the membership to be kept secret from sender.
  - Can be cheaper if recipients are geographically clustered around the list maintaining site.
  - More efficient if list size is bigger than message size.
  - Faster when distribution lists are contained in distribution lists.

Mail Handling

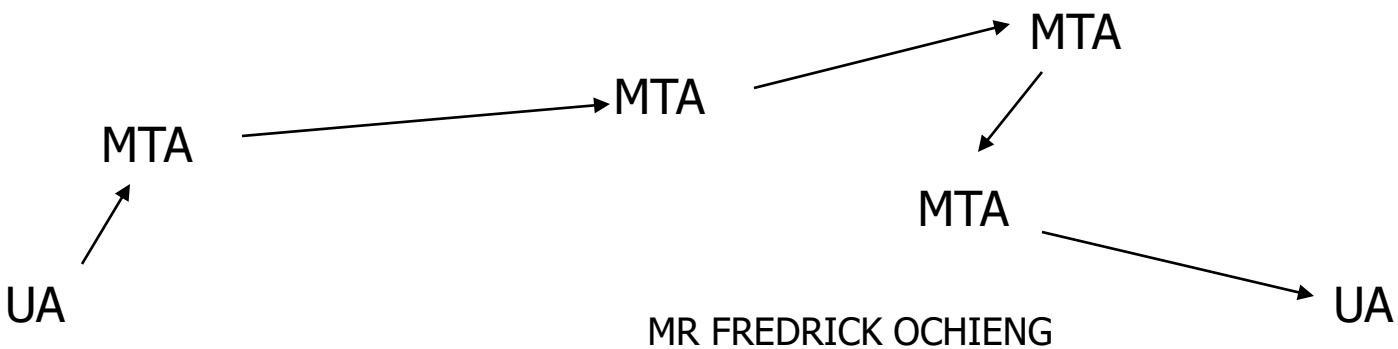
- Simplest: Send message directly from sender's machine to recipient's machine.
  - Works only if the recipients machine is always on.
  - Need **Electronic Post Boxes.**
    - Send mail to a machine permanently connected.

Mail Infrastructure

- Two Standards
  - X.400 family of protocols
    - Defined by International Telecommunications Union ITU and International Standardization Organization ISO
  - SMTP
    - Simple Mail Transfer Protocol
    - Defined by the Internet Engineering Task Force IETF.

Mail Infrastructure

- Mail infrastructure consists of a mesh of mail forwarders.
  - Called Message Transfer Agents (MTA)
  - Processing at source and destination done by User Agent (UA)



Mail Infrastructure

- Typically more than one path.
  - Deals with intermittent connections.
  - MTA could insist on authentication.
  - Security gateways through which all company mail is forwarded.
- Routing typically done manually.

Email Security Services

- Privacy
  - Keep anyone but the recipient from reading the message.
- Authentication
  - Receiver is reassured of the identity of the sender.
- Integrity
  - Receiver is reassured that the message has not been altered since transmission by sender.

Email Security Services

- Non-repudiation
  - Ability of recipient to prove (to a third party) that the sender really did send this message.
  - A.k.a. third party authentication.

Email Security Services

- Proof of submission
  - Verification given to the sender that the message was handed to the mail delivery system.
  - Not the same as a receipt by recipient / proof of delivery.
  - Possible to prove the identity of the message.

Email Security Services

- Proof of Delivery
  - Verification given to the sender that the message was handed to the UA of the recipient.
  - Not the same as proof of submission.
  - Possible to prove the identity of the message.

Email Security Services

- Message flow confidentiality.
  - Third party cannot tell whether email is exchanged between sender and recipient.
- Anonymity
  - The ability to send a message so that the receiver cannot tell the identity of the recipient.

Email Security Services

- Containment
  - Ability of the network to keep security levels of information from leaking out of a particular region.
- Audit
  - Capacity to log security relevant events.
- Accounting
  - Capacity to maintain system usage statistics and charge individual users.

Email Security Services

- Self Destruct
  - User should not be capable of forwarding or storing the message.
- Message Sequence Integrity
  - Reassurance that an entire sequence of messages arrived in the order transmitted and without losses.

Key Establishments

- Establishing Public Keys:
  - Out-of-band transmission
    - PGP public key hash on business card.
  - PKI
  - Piggy-backing of certificates on email messages.
- Establishing Secret Keys
  - Out-of-band transmission
  - Ticket via KDC.
    - Alice would obtain a ticket for Bob and attach it to her message to him.

Privacy

- Threats
  - Eavesdropping.
  - Relay nodes might store messages.
    - In fact, many relay nodes log messages completely.
  - Fundamentally, at sender and receiver's machine.
    - Email there is not in transit and not protected by the Electronic Communications Privacy Act.

Privacy

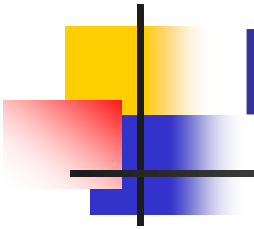
- End-to-End Privacy
  - Sender and recipient use encryption.
  - Complicated by multiple recipients.
  - Keys should be only used sparingly to avoid cipher attacks.
    - Alice chooses a secret key S.
    - Alice encrypts S with the key she shares with each recipient.

To: Bob, Carol, Dexter

From: Alice

Key-info: Bob 98932472138, Carol 129834298732, Dexter 100231098432

Message: qewroiu3219087v90(87sdh32198y*&97slknseiahfusdfiu39587(*



Privacy

- With Distribution List Exploders
  - Remote exploding:
    - Alice chooses a secret key S and encodes her message.
    - Alice attaches S encrypted to all recipients.
    - Distribution list exploder decodes S and attaches it encrypted to all recipients.
      - Remote exploder knows the contents.
  - Local exploding:
    - Alice needs to exchange keys with all people on the list.

Source Authentication

- With Public Key Technology
  - Alice can sign a message to Bob
    - By encrypting the whole message with her private key.
      - Then Bob would have to know Alice's public key.
      - Alice could embed her public key in the message together with a certificate or certificate chain.
    - By calculating a hash (MD5) of the message and encrypting it with her private key.
      - Then Bob does not need to know Alice's public key to read the mail.

Source Authentication

- With secret key technology
  - Alice and Bob share a secret S.
  - She can prove her identity by performing a computation on the message using S.
  - Result called
    - MIC – Message Integrity Code
    - MAC – Message Authentication Code.
  - Various methods:
    - MAC is the CBC residue of the message encrypted with S.
    - MAC is the encryption of the MD5 of message.
      - Then Alice only needs to repeat the encryption for various recipients.

Source Authentication

- With Distribution Lists
  - Public Keys: Easy.
    - Anyone can get Alice's public key.
  - Secret Keys: Hard.
    - Alice needs to share a key with the distribution list exploder.
    - Exploder will have to recalculate authentication data.
    - E.g. recalculate the encrypted hash with the recipients key.

Message Integrity

- Without Source Authentication
  - Why?
- With Source Authentication
  - Included if we calculate the authenticator from the complete message.

Repudiation

- Repudiation = Act of denying that a message was sent.
- Public Key Technology
  - Alice signs with her private key.
  - Bob can prove that Alice signed it.
- Hence non-repudiation.

  - Alice picks secret key S.
  - She encrypts S with Bob's public key: $\{S\}_{Bob}$.
  - She signs $\{S\}_{Bob}$ with her private key: $[\{S\}_{Bob}]_{Alice}$
  - She uses S to compute a MAC for the message.
  - She sends the message, the MAC, and $[\{S\}_{Bob}]_{Alice}$ to Bob.
- Bob knows that the message came from Alice because of Alice's private key.
- Bob can create any other message with S, therefore, he cannot prove that Alice send him that particular message.
- Hence repudiation.

Repudiation

- Secret Key Technology with non-repudiation
  - Needs a **notary** N.
    - Alice sends message to Bob first to N with source authentication.
    - Notary creates a **seal**.
      - Seal is something based on the message and Alice's name with a secret key that N does not share.
        - For example, encryption of message digest and Alice's name.
    - Bob needs to be able to verify the seal.
      - If Bob and N share a key, then N could verify the seal by sending an encryption of the message digest, Alice's name, and the seal.
      - Bob asks N to verify the seal.
    - Bob can prove that Alice sent this message.
  - Hence non-repudiation.

Proof of Submission / Delivery

- Email system can generate proof of receiving a message at any way station.
  - By handing out “seals” of sent messages.

Message Flow Confidentiality

- Needs an intermediary.
- Alice sends her email to Ivy, who forwards it to Bob.
- Alice periodically sends fake messages to Ivy.
- Ivy periodically sends fake messages to random recipients.

Anonymity

- Needs anonymity server.
- Freely available, but have difficulty with business model.

Containment

- Partition network into pieces that are capable of handling security classes.
- Mark each message with its security classification.
- Routers honor security rules.
  - I.e., refuse to forward to parts of a network not cleared for the security class of the message.

Text Formatting Issues

- No canonical text format
  - RFC 822 provides one format with <cr><lf> characters to separate lines.
    - But only works with SMTP.
  - Some mail servers remove white space at the end of lines, add line breaks to lines that are too long, etc.
  - This can break hashes and other MACs
- Data needs to be disguised as text.
  - uuencode
    - Uses 64 safe characters.
    - Data is encoded in these 64 characters
      - 6 bits encoded in 8 bits
  - S/MIME, PEM, PGP do something similar
    - The result is not readable by humans.

Verifying dates

- Preventing Backdating
  - Use a notary to verify messages.
    - Calculate MD5 of received message.
    - Send MD5 to notary.
    - Notary creates an encryption of MD5 and date.
    - Can include certificates used to establish sender's identity.
- Preventing Postdating
  - Include something in the message that you could only have known at the time that the message was sent.

Privacy Enhanced Mail: PEM

- Described in RFC 1421, 1422, 1423, 1424.
- Pretty much dead now.

Privacy Enhanced Mail: PEM

- PEM is implemented in software at the sender and the receiver, not in-between.
- PEM messages need to pass unchanged through mail-servers.
- PEM provides integrity protection and encryption

Privacy Enhanced Mail: PEM

- PEM message
  - Can consists of several blocks.
  - PEM flags them as separate, treated blocks.
    - Ordinary, unsecured data.
    - Integrity protected, unmodified data
    - Integrity protected encoded data
      - Encoded = safe to transmit through all mailers
    - Integrity protected, encoded, and encrypted data

Privacy Enhanced Mail: PEM

- Establishing keys
  - Per message key (random number)
  - Interchange key (public key)
    - To encrypt message key.

Privacy Enhanced Mail: PEM

- PEM Certificate Hierarchy
  - Single root CA (certification authority)
    - Internet Policy Registration Authority
      - Managed by the internet society
  - Public Certification Authorities
    - PCAs have different assurance levels.
    - There is only one path from the root CA to an individual

Privacy Enhanced Mail: PEM

- Certification
  - PEM allows Alice to send Bob her relevant certificates by including them in the PEM header.
- Certification Revocation Lists
  - Not included in header, hence
    - Two message types
      - CRL-Retrieval-Request to CRL service
      - CRL

Privacy Enhanced Mail: PEM

- Data canonicalization
  - How to get data through mail forwarders?
  - PEM encodes 6 bits into an 8b character

Privacy Enhanced Mail: PEM

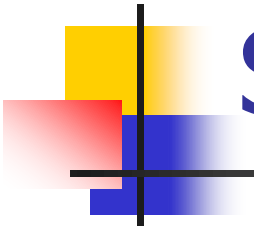
- Encryption
  - CBC mode with randomly chosen 64b initializing vector
    - To make exhaustive attacks against message keys more difficult.
  - Per message key distributed in header
  - Protected by interchange key.

Privacy Enhanced Mail: PEM

- Integrity protection
  - Message integrity code
    - MD2
    - MD5
  - Protected by cryptography
    - Alice signs the MIC with her private key.
      - When message is encrypted, the signed MIC needs to be encrypted as well.
    - Alice encrypts the MIC with the interchange key.

Privacy Enhanced Mail: PEM

- Multiple recipients
  - No problem for signed messages.
  - Encrypted messages are encrypted with the same key.
  - The per-message key is encrypted for each recipient individually.
- Forwarding
  - Should allow recipient to verify the signature of the original sender.
  - Only works with public keys.
    - If only integrity protected, only forwarding is necessary.
    - If encrypted, first receiver decrypts the per-message key, reencrypts it with the final receivers public key, and forwards.



S/MIME

- RFC 822 defines format for text messages sent using e-mail
- MIME deals with shortcomings.
  - SMTP cannot transmit executable or other binary data. (But UNIX users can use uuencode / uudecode.)
  - SMTP text is confined to the 128 7-bit ASCII characters.
  - SMTP servers can limit the size of email.
  - Not all SMTP implementations adhere completely to the SMTP standard. Problems are with the treatment of carriage return and linefeeds, truncating or wrapping lines longer than 76 characters, padding lines, and conversion of tab characters.

S/MIME

- Mime introduces five new headers:
  - MIME-Version:
    - This field must have a parameters value of 1.0 to indicate that the message conforms to RFC 2045 and 2046.
  - Content-Type:
    - Describes the data contained in the body so that the receiver can pick the appropriate application to represent the data to the user.
  - Content-Transfer-Encoding:
    - Indicates the type of transformation that has been used to represent the body of the message in order to render it amenable to the mail tranport.
  - Content-ID:
    - Used to identify MIME entities.
  - Content-Description:
    - A text description of the object within the body, e.g. audio-data.

S/MIME

- Multipart content-type
  - Content type header contains a parameter that defines the delimiter between body parts.

From: Nathaniel Borenstein <nsb@bellcore.com>
To: Ned Freed <ned@innosoft.com>
Date: Sun, 21 Mar 1993 23:56:48 -0800 (PST)
Subject: Sample message
MIME-Version: 1.0
Content-type: multipart/mixed; boundary="simple boundary" This is the preamble. It is to be ignored, though it
is a handy place for composition agents to include an
explanatory note to non-MIME conformant readers.

--simple boundary

This is implicitly typed plain US-ASCII text.
It does NOT end with a linebreak.
--simple boundary
Content-type: text/plain; charset=us-ascii
This is explicitly typed plain US-ASCII text.
It DOES end with a linebreak.

--simple boundary--
This is the epilogue. It is also to be ignored.

S/MIME

- MIME transfer encodings
  - 7bit (7b ASCII characters)
  - 8bit (uses the full ASCII character set),
  - 7bit (7b ASCII characters), 8bit (uses the full ASCII character set), binary, quoted-printable (the data is encoded as mostly ASCII text and remains readable by humans), and base64 (encodes 6-bit blocks of input to 8-bit blocks of output, all of which are printable ASCII characters). 7bit (7b ASCII characters), 8bit (uses the full ASCII character set), binary, quoted-printable (the data is encoded as mostly ASCII text and remains readable by humans), and base64 (encodes 6-bit blocks of input to 8-bit blocks of output, all of which are printable ASCII characters). 7bit (7b ASCII characters), 8bit (uses the full ASCII character set), binary, quoted-printable (the data is encoded as mostly ASCII text and remains readable by humans), and base64 (encodes 6-bit blocks of input to 8-bit blocks of output, all of which are printable ASCII characters). binary,
  - quoted-printable (the data is encoded as mostly ASCII text and remains readable by humans),
  - base64 (encodes 6-bit blocks of input to 8-bit blocks of output, all of which are printable ASCII characters).

S/MIME

- S/MIME provides
  - **Enveloped Data**
    - to apply privacy protection to a message. A sender needs to have access to a public key for each intended message recipient.
  - **Signed Data**
    - to provide authentication. Only a S/MIME enabled mailer can view this message.
  - **Clear-signed Data**
    - to provide authentication for users with S/MIME capabilities, but to retain readability other viewers.
  - Nesting of signed and encrypted data.

S/MIME

- S/MIME incorporates three public-key algorithms
  - DSS for digital sigantures
  - Diffie-Hellman for encyrpting session keys
  - RSA.
- SHA1 or MD5 for calculating digests
- Three-key triple DES for message encryption.
  - In an ideal situation, a S/MIME sender has a list of preferred decrypting capabilities from an intended recipient, in which case it chooses the best encryption. Otherwise, if the sender has received any previous mail from the intended recipient, it then chooses the same encryption mechanism.

S/MIME

- To secure a MIME entity
  - e.g. the entire message with exception of the RFC 822 header
  - S/MIME produces a PKCS object.
    - Cryptographic Token Interface Standard
  - PKCS object is then treated as the message object and encoded with MIME.
    - Since the result of encryption is typically in binary, it needs to be transferred in a more secure way, such as in base64 mode.

S/MIME

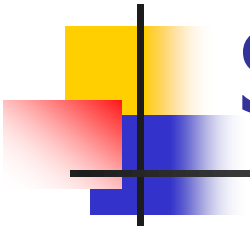
- To make an EnvelopedData MIME entity
  - generate a pseudo-random session key for either TripleDES or RC2/40 (a weak, exportable encryption).
  - for each recipient, encrypt the session key with the recipients public RSA key.
  - for each recipient, prepare a block known as RecipientInfo that contains the sender's public-key certificate, an identifier for the algorithm used to encrypt the session key, and the encrypted session key.
  - encrypt the message content with the session key.
- To recover the encrypted message, the recipient first reconverts the base64 encoding and uses his private key to recover the session key. He uses this key to decrypt the message.

S/MIME

- To make an SignedData MIME
  - select either SHA1 or MD5
  - compute the message digest of the content to be signed
  - encrypt the message digest with the signer's private key
  - prepare the SignerIno block that contains the signer's public key certificate, an identifier of the message digest algorithm, an identifier of the algorithm used to encrypt the message digest, and the encrypted message digest.
  - the whole block is then encoded in to base64 (excluding the RFC 822 header).

S/MIME

- Clear signing
  - Uses multipart content type in MIME to transmit body and signature separately.
  - Body needs to be encoded.
  - Signature is sent in base64.



S/MIME

- Certification Hierarchy
  - No particular public key infrastructure prescribed.
    - Public certifier (Versign, Thawte)
    - Organizational certifier
    - Certificates from any CA.

Pretty Good Privacy

- More than just a mail protocol.
- Interesting history.
  - "Guerilla Freeware"
- Number of incompatible versions

PGP: Pretty Good Privacy

- PGP uses public key cryptography.
  - Anarchic certificate model:
    - Everybody issues certificates and forwards public keys.
    - Users decide on trust rules.
  - Elaborate system of generating public-private keys.
  - Data on public keys, certificates, and people is combined in a key ring.
    - Key rings can be exchanged to build up trust databases.

PGP: Pretty Good Privacy

- Transfer Encoding
  - User specifies type of file for handling
    - Binary
    - Text file
  - Binary files are encoded at most once in order to prepare them for mail transit.
  - All files are compressed.

PGP: Pretty Good Privacy

- PGP messages
  - PGP uses IDEA.
  - Message is prefaced with the IDEA key encrypted with the recipients public key.