IA 221: Network Security

Lecture 2: Application Layer Security

Application Layer

- Provides services for an application to send and recieve data over the network, e.g., telnet (port 23), mail (port 25), finger (port 79)

- Interface to the transport layer
  - Operating system dependent
  - Socket interface

Providing Security

- Provide security system that can be used by different applications
  - Develop authentication and key distribution models
- Enhance application protocol with security features
  - Need to enhance each application

Authentication and Key Distribution

- Kerberos (MIT) and its extensions (Secure European System for Application in a Multi-vendor Environment (SESAME))
- Network Security Program (IBM)
- SPX (Digital Equipment Corporation)
- The Exponential Security System (University of Karlsruhe)

Kerberos Threats

- User gains access to workstation and pretends to be another user operating from that workstation
- User may alter the network address of a workstation so that the requests form the altered workstation appear to come from the impersonated workstation.
- User may eavesdrop on exchanges and use a replay attack to gain access to a server or to disrupt operation.

Requirements

- Secure
- Reliable
- Transparent
- Scalable

- Trusted Third Party authentication service
- Based on Needham-Schroeder (1978) protocol

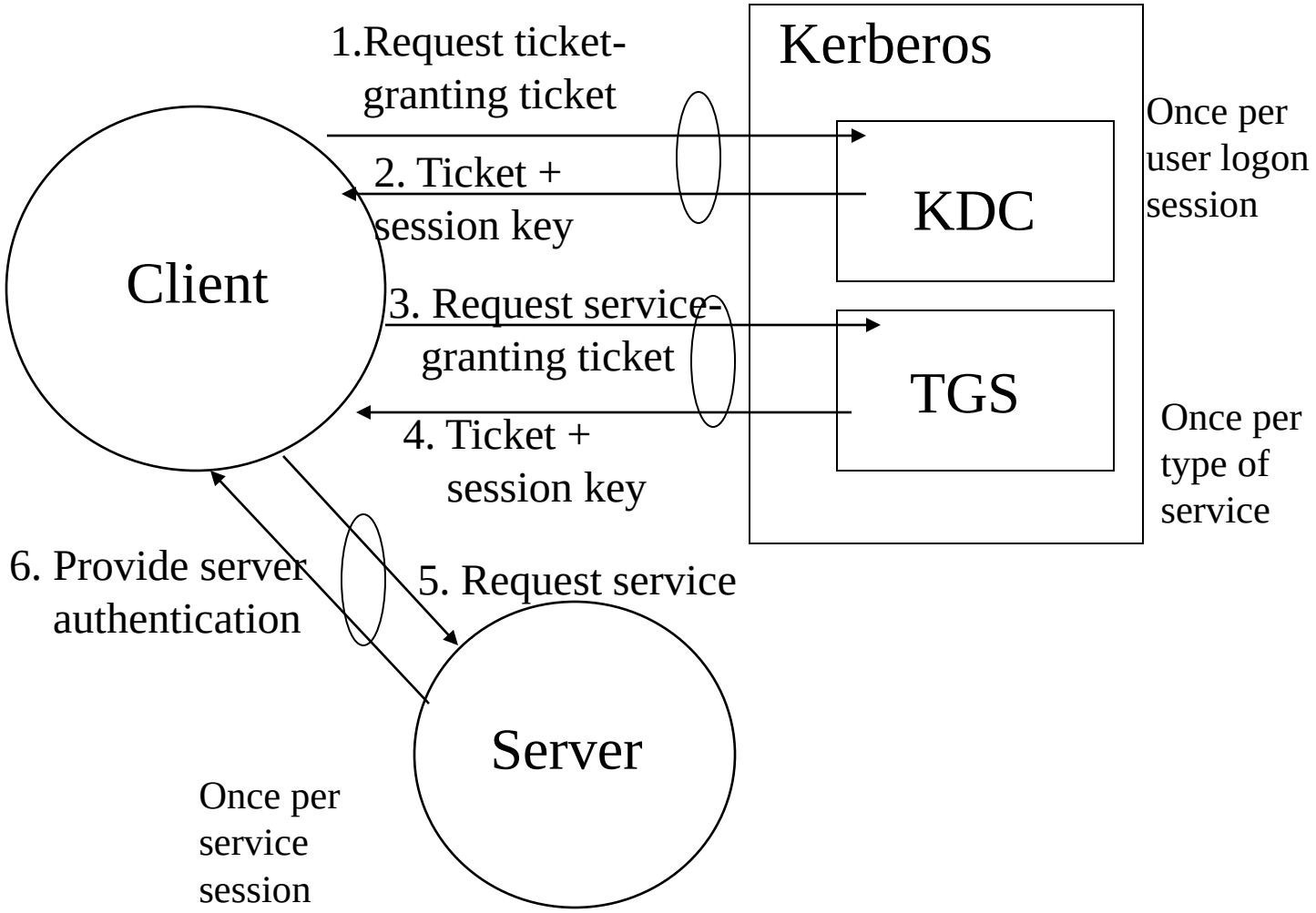
Kerberos Components

- Key Distribution Center (KDC)
  - Authentication server (AS)
  - Ticket-granting server (TGS)
- Database: users' identifiers + secret kay shared between KDC and user
- Need physical security

Ticketing System

- KDC issues tickets that clients and servers can use to mutually authenticate themselves and agree on shared secrets.
- Ticket:
  - Session key
  - Name of principal
  - Expiration time
- Ticket types:
  - Ticket-granting ticket: issued by AS and used between client and TGS
  - Service ticket: issued by TGS and used between client and server

Kerberos


Kerberos
1.Request ticket-granting ticket
2. Ticket + session key
Client
KDC
3. Request service-granting ticket
TGS
4. Ticket + session key
Once per user logon session
Once per type of service
6. Provide server authentication
5. Request service
Server
Once per service session

Kerberos Versions

- Version 4 (MIT) – 1992
  - Versions 1-3 were only used at MIT
  - Shortcomings and limited functionality (S. Bellovin and M. Merrit 1990)
- Version 5 (RFC 1510) – 1993
  - Improves on version 4 shortcomings

Version 4 limitations

- Environmental shortcomings
  - Encryption system dependence
  - Internet protocol dependence
  - Message byte ordering
  - Ticket lifetime
  - Authentication forwarding
  - Inter-realm authentication
- Technical deficiencies
  - Double encryption
  - PCBC encryption
  - Session keys
  - Password attack

Security-Enhanced Application Protocol

- Applications:
  - Terminal access
  - File transfer
  - Electronic mail
  - WWW transactions
  - DNS
  - Distributed file system

Terminal Access

- Protocols running on top of TCP/IP
  - Telnet: password based authentication
  - Rlogin: address-based authentication
- Security enhanced Telnet
  - Kerberos-mediated Telnet encryption: difficult to achieve
  - Security-enhanced Telnet (e.g., Secure Telnet (STEL) Univ. Milan
    - Authentication enforced by STEL is stronger than Telnet
    - All data traffic is encrypted between client and server
- Secure Shell (SSH)

SSH

- Provides similar services than SSL
  - Mutual authentication
  - Encrypted sessions between two endpoints
- Most often used to replace traditional terminal access → Application layer security
- Any application running on top of TCP can be secured by SSH

SSH versions

- SSH v1
  - Tatu Ylonen, Helsinki University of Technology, Finland
  - Implementation, source code, documentation, configuration scripts: public and freely available
  - Widespread use
- SSH v2
  - Specified by IETF Secure Shell WG (1st draft: 1997)
  - Widespread use
- Open source implementations: OpenSSH

SSH

- Both version use generic transport layer security protocol over TCP/IP
- Support for
  - Host and user authentication
  - Data compression
  - Data confidentiality
  - Integrity protection
- Server listens for TCP connection on port 22, assigned to SSH

SSH v1 keys

- Host public key pair
  - Bind connection to the desired server host
  - Long-term
  - Long key size (typically 1,024 bit RSA)
- Server public key pair
  - Provide confidentiality
  - Short-term
  - Short key size (typically 768 bit RSA)
  - Changes periodically (i.e., every hour by default)
  - For PFS server’s private key cannot be saved on disk

SSH Session

- Client → Server: Authentication request
- Server → Client: Server public keys (long-term and short-term)
- Client:
  - Compares received keys to its database of pre-distributed keys and (usually) accepts keys
  - Generates 256-bit random session key
  - Chooses encrypting algorithm
  - Pads session key
  - Double encrypts session key with server and host public keys

SSH Session

- Client → Server: Sends double encrypted session key
- Server:
  - Decrypts session key
- Server → Client: send confirmation encrypted by session key

Both parties use session key to encrypt traffic between server and client

Authentication

- After session key agreement, client assumes that the server is authenticated
- If user authentication is required:
  - Password authentication
  - RSA authentication (server need to know the user's public key)

Electronic Mail Security

Electronic Mail

- Most heavily used network-based application
- Used across different architectures and platforms
- Send e-mail to others connected directly or indirectly to the Internet regardless of host operating systems and communication protocols
- NEED:
  - Authentication
  - Confidentiality

Secure E-mail Approaches

- PGP: Pretty good Privacy
- PEM: Privacy-Enhanced Mail
- Secure Multipurpose Internet Mail Extensions (S/MIME)

Pretty Good Privacy

- Phil Zimmermann
- Confidentiality and authentication for
  - Electronic mail and
  - Storage applications

PGP – Evolution

1. Choose best available cryptographic algorithms
2. Integrate these algorithms such that
   - ❖ Independent of operating system and processor
   - ❖ Based on a small set of commands
3. Make the application and the documentation available through the Internet
4. Create an agreement with a company to provide compatible, low-cost commercial version of PGP

PGP - Usage

PGP became widely used within a few years

- Available worldwide for different platforms
- Based on proven secure algorithms (RSA, IDEA, MD5)
- Wide range of applicability
- Was not developed or controlled by government standards

Why PGP?

- Protect privacy
  - “I don’t need encryption!” = “I don’t need privacy.”
    - Interception transmission to destinations
    - Transparent mailbox (dial-up connection)
    - You may not but other party may want privacy
  - Commercial privacy
    - Customer’s data
    - Company data
  - User’s profiling
- Signed messages
  - Authentication
  - Integrity

?

PGP Services

- **Digital Signature**: RSA, MD5
  - Hash code of message is created using MD5, encrypted using RSA, with sender's private key, and attached to the message
- **Confidentiality**: RSA, IDEA
  - Message is encrypted using IDEA, with one-time session key generated by the sender, session key is encrypted, using RSA and the recipient's public key, and attached to the message

PGP Services

- **Compression**: ZIP
  - Message may be compressed for storage or transmission
- **E-mail compatibility**: Radix 64 conversion
  - Encrypted message is converted to ACSII string
- **Segmentation**
  - To accommodate maximum message size, PGP performs segmentation and reassembly

Authentication

- RSA and MD5: effective digital signature schema
- Default: signatures attached to messages
- Support detached signatures
  - User may want to maintain separate signature files
  - Detect virus infection of executable programs
  - Support multiple signature of a message

Confidentiality

- IDEA: secret-key encryption
- Key-distribution:
  - randomly generated, one-time session keys
  - Encrypted by receiver's public key
  - Attached to the message
- Double encryption
  - IDEA
  - One-time key
- RSA key size:
  - Casual: 384 bits
  - Commercial: 512 bits
  - Military: 1024 bits

Compression

- Usually after signature and before encryption
  - Preferable to sign uncompressed message -> store them together for future verification
  - PGP’s compression algorithm is not deterministic
  - Encryption after compression strengthen cryptographic security (less redundancy)

E-mail Compatibility

- PGP encryption: arbitrary 8-bit binary stream
- Several e-mail system: ASCII text
- PGP: converts the binary stream to a stream of printable ASCII characters
  - Expands the message by 33%
  - Converts everything, regardless of content (even ASCII characters)

Segmentation and Reassembly

- E-mail: restriction on maximum message length
  - Long messages broken into segments
  - Segments are mailed separately
- PGP automatically divides a long message
- Segmentation is done after all other processing
- Receiving PGP reassembles the original message

S/MIME (Secure Multipurpose Internet Mail Extension)

- Created to deal with short comings of PGP
  - Support for multiple formats in a message, not just ASCII text
  - Support for IMAP (Internet Mail Access Protocol)
  - Support for multimedia
- Similar to PGP, can also do authentication, encryption, or both
- Use X.509 PKI and public-key certificates
- Also support standard symmetric-key encryption, public-key encryption, digital signature algorithms, hash functions, and compression functions

S/MIME Functions

- S/MIME content-types support four new functions:
- **Enveloped data:** This consists of encrypted content of any type and encrypted-content encryption keys for one or more recipients.
- **Signed data:** A digital signature is formed by taking the message digest of the content to be signed and then encrypting that with the private key of the signer. The content plus signature are then encoded using base64 encoding. A signed data message can only be viewed by a recipient with S/MIME capability.
- **Clear-signed data:** As with signed data, a digital signature of the content is formed. However, in this case, only the digital signature is encoded using base64. As a result, recipients without S/MIME capability can view the message content, although they cannot verify the signature.
- **Signed and enveloped data:** Signed-only and encrypted-only entities may be nested, so that encrypted data may be signed and signed data or clear-signed data may be encrypted.

S/MIME Cryptographic Algorithms

- Default algorithms used for *signing* messages are Digital Signature Standard (**DSS**) and **SHA-1**
- RSA public-key encryption algorithm can be used with **SHA-1** or the **MD5** *message digest* algorithm for forming signatures
- **Radix-64** or **base64** mapping is used to *map* the signature and message *into printable ASCII* characters

S/MIME Public Key Certificates

- Default algorithms used for encrypting S/MIME message are **3DES** and **EI-Gamal**
  - EI-Gamal is based on the Diffie-Hellman public-key exchange algorithm
- If encryption is used alone **radix-64** is used to convert the ciphertext to ASCII format
- Basic tool that permits widespread use of S/MIME is the public-key certificate
  - S/MIME uses certificates that conform to the international standard X.509v3

PEM (Privacy-Enhanced Mail)

- A draft Internet Standard (1993).
- Used with SMTP.
- Implemented at application layer.
- Provides:
  - Disclosure protection
  - Originator authenticity
  - Message integrity

Summary of PEM Services

Function	Algorithms used	Description
Message Encryption using RSA	DES	A message is encrypted using DES-CBC. The session key is encrypted with the recipient's public key.
Authentication and Digital signature (asymmetric encryption)	RSA with MD2 or MD5	A hash code of a message is created using MD2 or MD5. This is encrypted using RSA with the sender's private key.
E-mail compatibility	Radix 64 conversion	To provide transparency for e-mail applications.

Secure-HTTP (S-HTTP)

- Secure HTTP (S-HTTP) extends the Hypertext Transfer Protocol (HTTP).
- Each S-HTTP file is either encrypted, contains a digital certificate, or both.
- S-HTTP design provides for secure communications, primarily commercial transactions, between a HTTP client and a server.
- It does this through a wide variety of mechanisms to provide for confidentiality, authentication, and integrity while separating policy from mechanism

Secure-HTTP (S-HTTP) …

- During the transfer transaction, both the client browser and the server, use the information contained in the HTTP header to negotiate formats they will use to transfer the requested information.
- The S-HTTP protocol extends this negotiation between the client browser and the server to include the negotiation for security matters.
- Hence S-HTTP uses additional headers for message encryption, digital certificates and authentication in the HTTP format which contains additional instructions on how to decrypt the message body.

Hypertext Transfer Protocol over Secure Socket Layer (HTTPS)

- HTTPS is the use of Secure Sockets Layer (SSL) as a sub-layer under the regular HTTP in the application layer.
- It is also referred to as Hypertext Transfer Protocol over Secure Socket Layer (HTTPS) or HTTP over SSL, in short.
- HTTPS is a Web protocol developed by Netscape, and it is built into its browser to encrypt and decrypt user page requests as well as the pages that are returned by the Web server.
- HTTPS uses port 443 instead of HTTP port 80 in its interactions with the lower layer, TCP/IP

Secure Electronic Transactions (SET)

- SET is a cryptographic protocol developed by a group of companies that included Visa, Microsoft, IBM, RSA, Netscape, MasterCard and others.
- It is a highly specialized system with complex specifications contained in three books with book one dealing with the business description, book two a programmer's guide, and book three giving the formal protocol description.
- For each transaction, SET provides the following services: authentication, confidentiality, message integrity, and linkage.
- SET uses public key encryption and signed certificates to establish the identity of every one involved in the transaction and to allow every correspondence between them to be private.