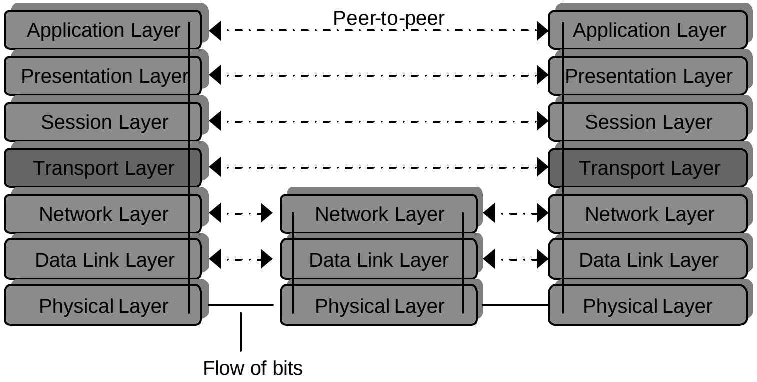
Network Security Assurance

Lecture 9

October 30, 2003

ISO/OSI Model
SSL: Security at Transport Layer



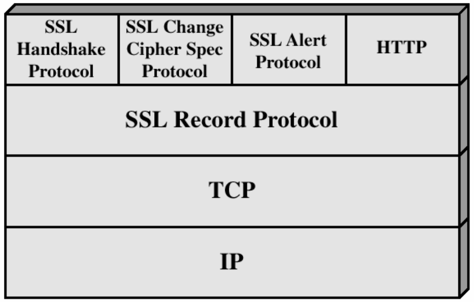
Security at the Transport Layer Secure Socket Layer (SSL)

- Developed by Netscape to provide security in WWW browsers and servers
- SSL is the basis for the Internet standard protocol – Transport Layer Security (TLS) protocol (compatible with SSLv3)
- Key idea: *Connections* and *Sessions*
  - A SSL session is an association between two peers
  - An SSL connection is the set of mechanisms used to transport data in an SSL session

Secure Socket Layer (SSL)

- Each party keeps session information
  - Session identifier (unique)
  - The peer's X.503(v3) certificate
  - Compression method used to reduce volume of data
  - Cipher specification (parameters for cipher and MAC)
  - Master secret of 48 bits
- Connection information
  - Random data for the server & client
  - Server and client keys (used for encryption)
  - Server and client MAC key
  - Initialization vector for the cipher, if needed
  - Server and client sequence numbers
- Provides a set of supported cryptographic mechanisms that are setup during negotiation (handshake protocol)

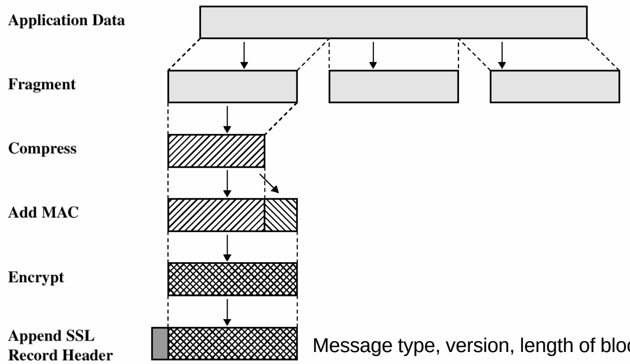
SSL Architecture



Provides a basis for Secure communication
Confidentiality +
Message authenticity

SSL Record Protocol Operation

e.g., HTTP messages



Message type, version, length of block

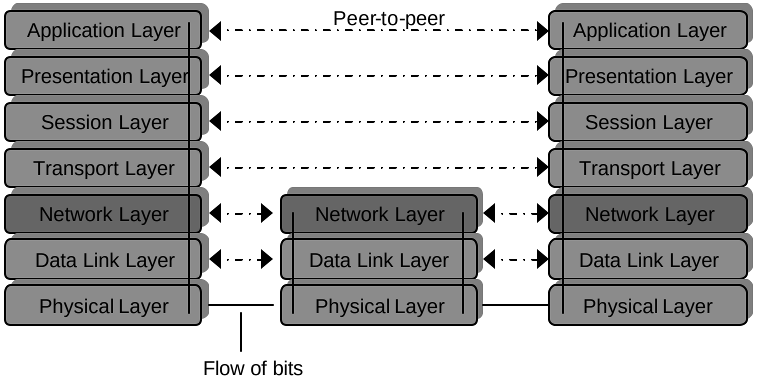
Handshake Protocol

- The most complex part of SSL
- Allows the server and client to authenticate each other
  - Based on interchange cryptosystem (e.g., RSA)
- Negotiate encryption, MAC algorithm and cryptographic keys
  - Four rounds
- Used before any application data are transmitted

Other protocols

- SSL Change Cipher Spec Protocol
  - A single byte is exchanged
  - After new cipher parameters have been negotiated (renegotiated)
- SSL Alert Protocol
  - Signals an unusual condition
  - *Closure alert* : sender will not send anymore
  - *Error alert*: fatal error results in disconnect

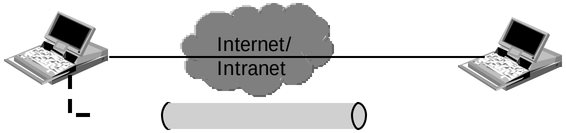
ISO/OSI Model
IPSec: Security at Network Layer



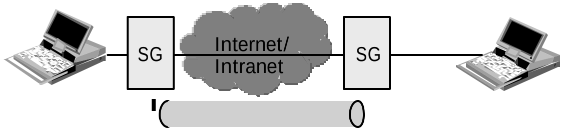
IPSec

- Set of protocols/mechanisms
  - Encrypts and authenticates all traffic at the IP level
    - Protects all messages sent along a path
    - Intermediate host with IPSec mechanism (firewall, gateway) is called a *security gateway*
  - Use on LANs, WANs, public, and private networks
- Application independent (Transparent to user)
  - Web browsing, telnet, ftp…
- Provides at the IP level
  - Access control
  - Connectionless integrity
  - Data origin authentication
  - Rejection of replayed packets
  - Data confidentiality
  - Limited traffic analysis confidentiality

Cases where IPSec can be used

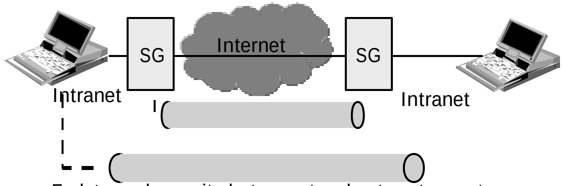


End-to-end security between two hosts

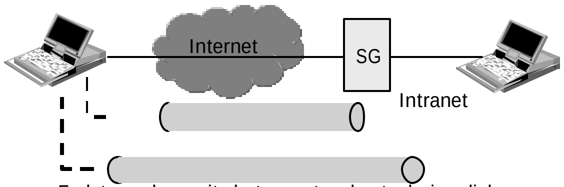


End-to-end security between two security gateways

Cases where IPSec can be used (2)



End-to-end security between two hosts + two gateways



End-to-end security between two hosts during dial-up

IPSec Protocols

- Authentication header (AH) protocol
  - Message integrity
  - Origin authentication
  - Anti-replay services
- Encapsulating security payload (ESP) protocol
  - Confidentiality
  - Message integrity
  - Origin authentication
  - Anti-replay services
- Internet Key Exchange (IKE)
  - Exchanging keys between entities that need to communicate over the Internet
  - What authentication methods to use, how long to use the keys, etc.

Security Association (SA)

- Unidirectional relationship between peers (a sender and a receiver)
- Specifies the security services provided to the traffic carried on the SA
  - Security enhancements to a channel along a path
- Identified by three parameters:
  - IP Destination Address
  - Security Protocol Identifier
    - Specifies whether AH or ESP is being used
  - Security Parameters Index (SPI)
    - Specifies the security parameters associated with the SA

Security Association (2)

- Each SA uses AH or ESP (not both)
  - If both required two are SAs are created
- Multiple security associations may be used to provide required security services
  - A sequence of security associations is called *SA bundle*
  - Example: We can have an AH protocol followed by ESP or vice versa

Security Association Databases

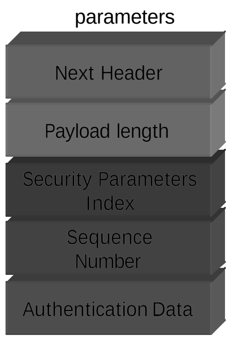
- IP needs to know the SAs that exist in order to provide security services
- Security Policy Database (SPD)
  - IPSec uses SPD to handle messages
  - For each IP packet, it decides whether an IPSec service is provided, bypassed, or if the packet is to be discarded
- Security Association Database (SAD)
  - Keeps track of the sequence number
  - AH information (keys, algorithms, lifetimes)
  - ESP information (keys, IVs, algorithms, lifetimes)
  - Lifetime of the SA
  - Protocol mode
  - MTU

IPSec Modes

- Two modes
  - Transport mode
    - Encapsulates IP packet data area
    - IP Header is not protected
      - Protection is provided for the upper layers
      - Usually used in host-to-host communications
  - Tunnel mode
    - Encapsulates entire IP packet in an IPSec envelope
      - Helps against traffic analysis
      - The original IP packet is untouched in the Internet

Authentication Header (AH)

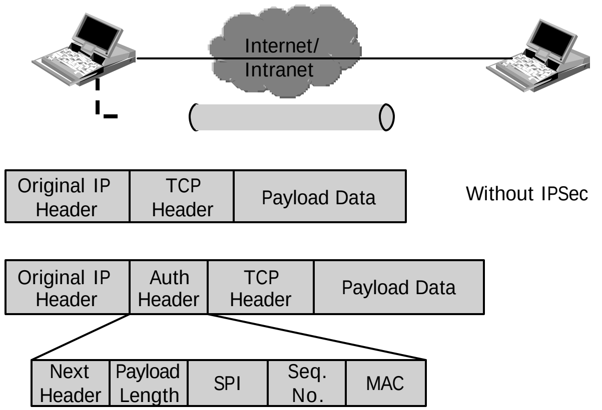
- Next header
  - Identifies what protocol header follows
- Payload length
  - Indicates the number of 32-bit words in the authentication header
- Security Parameters Index
  - Specifies to the receiver the algorithms, type of keys, and lifetime of the keys used
- Sequence number
  - Counter that increases with each IP packet sent from the same host to the same destination and SA
- Authentication Data



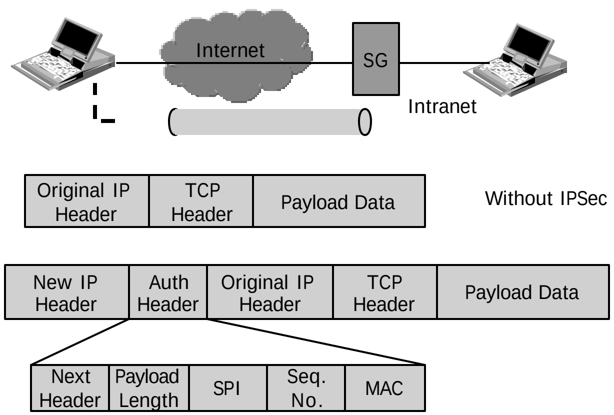
Preventing replay

- Using 32 bit sequence numbers helps detect replay of IP packets
- The sender initializes a sequence number for every SA
  - Each succeeding IP packet within a SA increments the sequence number
- Receiver implements a window size of W to keep track of authenticated packets
- Receiver checks the MAC to see if the packet is authentic

Transport Mode AH

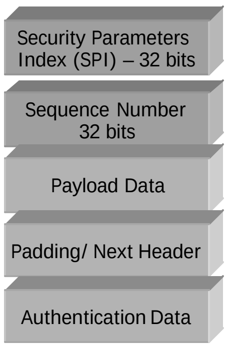


Tunnel Mode AH



ESP – Encapsulating Security Payload

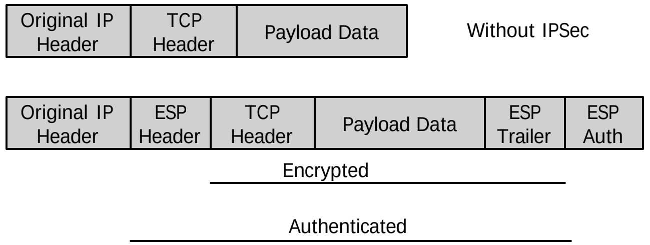
- Creates a new header in addition to the IP header
- Creates a new trailer
- Encrypts the payload data
- Authenticates the security association
- Prevents replay



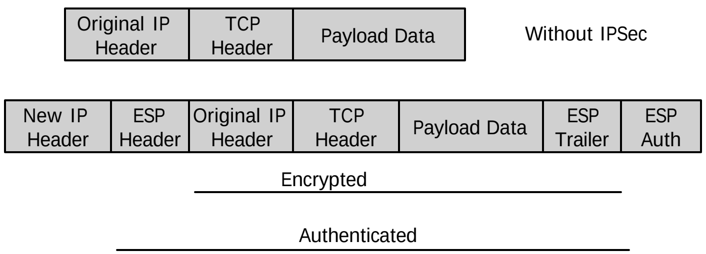
Details of ESP

- Security Parameters Index (SPI)
  - Specifies to the receiver the algorithms, type of keys, and lifetime of the keys used
- Sequence number
  - Counter that increases with each IP packet sent from the same host to the same destination and SA
- Payload
  - Application data carried in the TCP segment
- Padding
  - 0 to 255 bytes of data to enable encryption algorithms to operate properly
  - To mislead sniffers from estimating the amount of data transmitted
- Authentication Data
  - MAC created over the packet

Transport mode ESP



Tunnel mode ESP



Perimeter Defense

- Organization system consists of a network of many host machines –
  - the system is as secure as the weakest link
- Use perimeter defense
  - Define a border and use gatekeeper (firewall)
- If host machines are scattered and need to use public network, use encryption
  - Virtual Private Networks (VPNs)

Perimeter Defense

- Is it adequate?
  - Locating and securing all perimeter points is quite difficult
    - Less effective for large border
  - Inspecting/ensuring that remote connections are adequately protected is difficult
  - Insiders attack is often the most damaging

Firewalls

- Total isolation of networked systems is undesirable
  - Use firewalls to achieve selective border control
- Firewall
  - Is a configuration of machines and software
  - Limits network access
  - Come "for free" inside many devices: routers, modems, wireless base stations etc.
  - Alternate:

    a firewall is a host that mediates access to a network, allowing and disallowing certain type of access based on a configured security policy

What Firewalls can't do

- They are not a panacea
  - Only adds to defense in depth
- If not managed properly
  - Can provide false sense of security
- Cannot prevent insider attack
- Firewalls act a particular layer (or layers)

Virtual Private Networks What is it?

- It is a private network that is configured within a public network
- A VPN "appears" to be a private national or international network to a customer
- The customer is actually "sharing" trunks and other physical infrastructure with other customers
- Security?

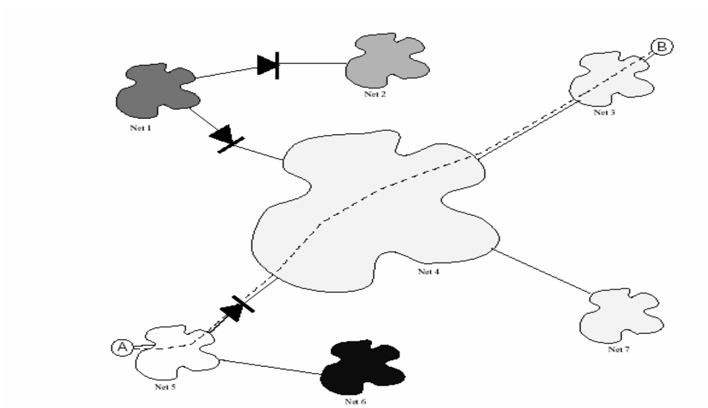
What is a VPN? (2)

- A network that supports a *closed* community of authorized users
- The authorized users are allowed to access various network related resources and services
- There is traffic isolation
  - Contents are secure
  - Services and resources are secure
- Use the public Internet as part of the virtual private network
- Provide security!
  - Confidentiality and integrity of data
  - User authentication
  - Network access control
- IPSec

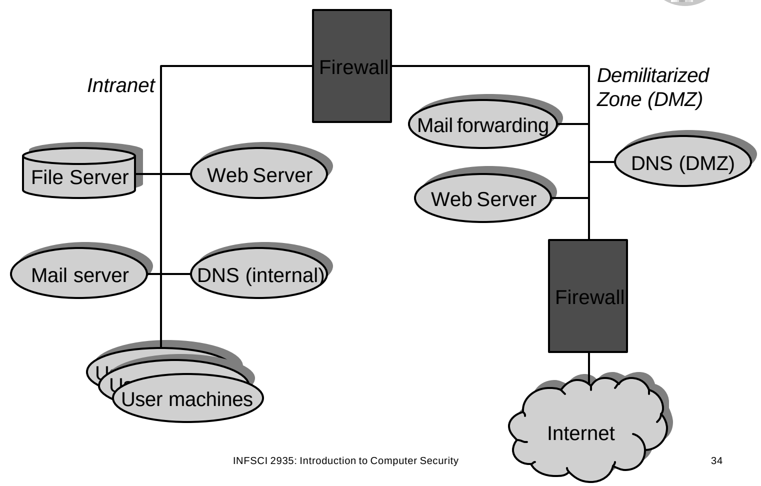
Secure IP VPNs

- Use the public Internet as part of the virtual private network
- Provide security!
  - Confidentiality and integrity of data
  - User authentication
  - Network access control
- IPSec can be used

Tunneling in VPN



"Typical" corporate network



Typical network: Terms

- Network Regions
  - Internet
  - Intranet
  - DMZ
- Network Boundaries
  - Firewall
    - Filtering firewall: Based on packet headers
    - Audit mechanism
  - Proxy
    - Proxy firewall: Gives external view that hides intranet
    - Contents of packets and messages besides attributes of packet headers

Issues

- IP: Intranet hidden from outside world
  - Internal addresses can be real
    - Proxy maps between real address and firewall
  - Fake private addresses
    - Network Address Translation protocol maps internal addresses to the Internet addresses (inner firewall)
- Mail Forwarding
  - Hide internal addresses
  - Map incoming mail to “real” server
  - Additional incoming/outgoing checks

Firewalls: Configuration

- External Firewall
  - What traffic allowed
    - External source: IP restrictions
    - What type of traffic: Ports (e.g., SMTP, HTTP)
  - Proxy between DMZ servers and internet
- Internal Firewall
  - Traffic restrictions: Ports, From/to IP
  - Proxy between intranet and outside

DMZ Administration

- Direct console access required?
  - Real hassle
- "Special" access
  - SSH connections allowed from internal to DMZ "administration" connections
  - Only from specified internal IPs
  - Only through internal firewall

Assurance

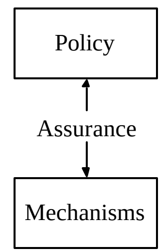
Overview

- Trust
- Problems from lack of assurance
- Types of assurance
- Life cycle and assurance
- Waterfall life cycle model
- Other life cycle models

Trust

- *Trustworthy* entity has sufficient credible evidence leading one to believe that the system will meet a set of requirements
- *Trust* is a measure of trustworthiness relying on the evidence
- *Assurance* is confidence that an entity meets its security requirements based on evidence provided by the application of assurance techniques
  - *Formal methods, design analysis, testing etc.*

Relationships



Statement of requirements that explicitly defines the security expectations of the mechanism(s)

Provides justification that the mechanism meets policy through assurance evidence and approvals based on evidence

Executable entities that are designed and implemented to meet the requirements of the policy

Evaluation standards
Trusted Computer System Evaluation Criteria
Information Technology Security Evaluation Criteria
Common Criteria

Problem Sources (Neumann)

1. Requirements definitions, omissions, and mistakes
2. System design flaws
3. Hardware implementation flaws, such as wiring and chip flaws
4. Software implementation errors, program bugs, and compiler bugs
5. System use and operation errors and inadvertent mistakes
6. Willful system misuse
7. Hardware, communication, or other equipment malfunction
8. Environmental problems, natural causes, and acts of God
9. Evolution, maintenance, faulty upgrades, and decommissions

Examples

- Challenger explosion (1986)
  - Sensors removed from booster rockets to meet accelerated launch schedule
- Deaths from faulty radiation therapy system
  - Hardware safety interlock removed
  - Flaws in software design
- Bell V22 Osprey crashes
  - Failure to correct for malfunctioning components; two faulty ones could outvote a third
- Intel 486 chip bug (trigonometric function)
  - Cost a lot of time and money

Role of Requirements

- *Requirements* are statements of goals that must be met
  - Vary from high-level, generic issues to low-level, concrete issues
- *Security objectives* are high-level security issues and business goals
- *Security requirements* are specific, concrete issues

Types of Assurance

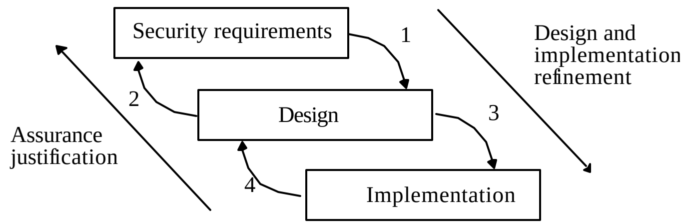
- *Policy assurance* is evidence establishing security requirements in policy is complete, consistent, technically sound
  - To counter threats and meet objectives
- *Design assurance* is evidence establishing design sufficient to meet requirements of security policy
- *Implementation assurance* is evidence establishing implementation consistent with security requirements of security policy
  - Need to use good engineering practices

Types of Assurance

- *Operational assurance* is evidence establishing system sustains the security policy requirements during installation, configuration, and day-to-day operation
  - Also called *administrative assurance*
  - Example,

    Do a thorough review of product or system documentation and procedures, to ensure that the system cannot accidentally be placed in a non-secure state.

Assurance steps



Life Cycle

- Conception
- Manufacture
- Deployment
- Fielded Product Life

Conception

- Idea
  - Decisions to pursue it
- Proof of concept
  - See if idea has merit
  - Rapid prototyping, analysis, etc.
- High-level requirements analysis
  - What does “secure” mean for this concept?
    - Identify threats
  - Is it possible for this concept to meet this meaning of security?
  - Is the organization willing to support the additional resources required to make this concept meet this meaning of security?

Manufacture

- Develop detailed plans for each group involved
  - May depend on use; internal product requires no sales
  - *Plans* : marketing, sales training, development, testing
  - Software development and engineering process
- Implement the plans to create entity
  - Includes decisions whether to proceed, for example due to market needs
- May be the longest stage

Deployment

- Delivery
  - Assure that correct (assured) masters are delivered to production and protected
  - Distribute to customers, sales organizations
- Installation and configuration
  - Developers must ensure that the system operates properly in the production environment

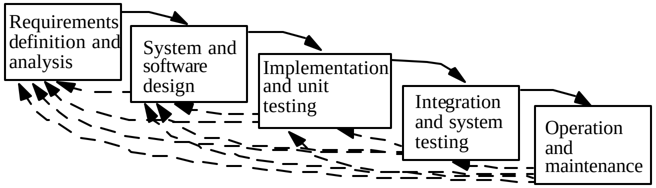
Fielded Product Life

- Routine maintenance, patching
  - Responsibility of engineering in small organizations
  - Responsibility may be in different group than one that manufactures product
- Customer service, support organizations
  - Answering questions; recording bugs
- Retirement or decommission of product
  - Migration plans for customers

Waterfall Life Cycle Model

- Requirements definition and analysis
  - Functional and non-functional
  - General (for customer), specifications
- System and software design
- Implementation and unit testing
- Integration and system testing
- Operation and maintenance

Relationship of Stages



Other Models of Software Development

- Exploratory programming
  - Develop working system quickly
  - Used when detailed requirements specification cannot be formulated in advance, and adequacy is goal
  - No requirements or design specification, so low assurance
- Prototyping (Similar to Exploratory)
  - Objective is to establish system requirements
  - Future iterations (after first) allow assurance techniques

Models

- Formal transformation
  - Create formal specification
  - Translate it into program using correctness-preserving transformations
  - Very conducive to assurance methods
- System assembly from reusable components
  - Depends on whether components are trusted
  - Must assure connections, composition as well
  - Very complex, difficult to assure
  - This is common approach to building secure and trusted systems

Models

- Extreme programming
  - Rapid prototyping and "best practices"
  - Project driven by business decisions
  - Requirements open until project complete
  - Programmers work in teams
  - Components tested, integrated several times a day
  - Objective is to get system into production as quickly as possible, then enhance it
  - Evidence adduced *after* development needed for assurance

Key Points

- Assurance critical for determining trustworthiness of systems
- Different levels of assurance, from informal evidence to rigorous mathematical evidence
- Assurance needed at all stages of system life cycle