email: Message integrity & authentication

- Alice wants to provide sender authentication message integrity. ...How?
- Alice digitally signs message
- Sends both message (in the clear) and digital signature

email: fully secure

- Alice wants to provide secrecy, sender authentication & message integrity. ...How?
- Alice uses three keys: her private key, Bob's public key, the newly created symmetric key
- What does Bob do to retrieve the msg & be sure it came from Alice?
Pretty good privacy (PGP)

- To Activity (to act out PGP)...
- Internet e-mail encryption scheme, de-facto standard.

Uses
- Symmetric key cryptography
- Public key cryptography
- Hash function
- Digital signature

Provides
- Secrecy
- Sender authentication
- Integrity

Chapter 8 roadmap

8.1 What is network security?
8.2 Principles of cryptography
8.3 Message integrity
8.4 Securing e-mail
8.5 Securing TCP connections: SSL
8.6 Network layer security: IPsec
8.7 Securing wireless LANs
8.8 Operational security: firewalls and IDS

Recall: Socket Programming

Application layer communication via the transport layer

goal: learn how to build client/server applications that communicate using sockets

socket: door between application process and end-to-end transport protocol

TCP Flow Chart
Recall: TCP Connection Management

**Connection Set Up:**

- **Step 1:** client sends TCP SYN segment
- **Step 2:** server receives SYN and replies with SYNACK
- **Step 3:** client receives SYNACK and replies with ACK

SSL: Secure Sockets Layer

- Provides
  - Confidentiality
  - Integrity
  - Authentication
- Available to all TCP applications
  - Secure socket interface

SSL: Secure Sockets Layer

SSl: Could be based on PGP

- But:
  - Want to send byte streams
  - Want certificate exchange to be part of protocol handshake phase
Basic SSL: a simple & secure channel

1. **Handshake**: Alice and Bob use their certificates and private keys to authenticate each other and exchange shared secret
2. **Key Derivation**: Alice and Bob use shared secret to derive set of keys
3. **Data Transfer**: Data to be transferred is broken up into a series of records
4. **Connection Closure**: Special messages to securely close connection

(1) SSL: Handshake

- Bob establishes TCP connection to Alice
- **Authenticates** Alice via CA signed certificate
- Creates, encrypts (using Alice’s public key), **sends master secret key** to Alice
  - (nonce exchange not shown)

(2) SSL: Key derivation

- Alice, Bob use **shared secret** (MS) to generate 4 keys:
  - $E_B$: Bob->Alice data encryption key
  - $E_A$: Alice->Bob data encryption key
  - $M_B$: Bob->Alice MAC key (the secret ‘bit pattern’)
  - $M_A$: Alice->Bob MAC key
- Encryption and MAC algorithms negotiable between hosts
- **Why 4 keys?**

(3) SSL: Data records

- Encrypt data in a constant stream as we write it to TCP? ... no!
- Where would we put the MAC?
- Instead, break stream into series of records
- Each record carries a MAC
- Receiver can act on each record as it arrives
(3) SSL: Data transfer

TCP byte stream block n bytes together

compute MAC

SSL record format

unencrypted encrypted using Eₐ

Real SSL: handshake (1)

Purpose
1. Server authentication
2. Negotiation: agree on crypto algorithms
3. Establish keys
4. Client authentication (optional)

Real SSL: handshake (2)
1. Client sends list of algorithms it supports, along with client nonce
2. Server chooses algorithms from list; sends back: choice + certificate + server nonce
3. Client verifies certificate, extracts server’s public key, generates pre_master_secret, encrypts with server’s public key, sends to server
4. Client and server independently compute encryption and MAC keys from pre_master_secret and nonces
5. Client sends a MAC of all the handshake messages
6. Server sends a MAC of all the handshake messages

SSL record protocol

record header: content type; version; length

MAC: includes sequence number, MAC key Mₛ

fragment: each SSL fragment 2¹⁴ bytes (~16 Kbytes)
Chapter 8 roadmap

8.1 What is network security?
8.2 Principles of cryptography
8.3 Message integrity
8.4 Securing e-mail
8.5 Securing TCP connections: SSL
8.6 Network layer security: IPsec
8.7 Securing wireless LANs
8.8 Operational security: firewalls and IDS

Firewalls

Firewalls: why

Prevent denial of service attacks:
- SYN flooding: attacker establishes many bogus TCP connections, no resources left for “real” connections

Prevent illegal modification/access of internal data
- e.g., attacker replaces CIA’s homepage with something else

Allow only authorized access to inside network
- set of authenticated users/hosts

Three types of firewalls:
- stateless packet filters
- stateful packet filters
- application gateways
Stateless packet filtering

- Internal network connected to Internet via router firewall
- Router filters packet-by-packet, decision to forward/drop packet based on:
  - Source IP address, destination IP address
  - TCP/UDP source and destination port numbers
  - ICMP message type
  - TCP SYN and ACK bits

Should arriving packet be allowed in? Departing packet let out?

- Example 1: block incoming and outgoing datagrams with IP protocol field = 17 and with either source or dest port = 23  
  - result: all incoming, outgoing UDP flows and telnet connections are blocked
- Example 2: block inbound TCP segments with ACK=0.  
  - result: prevents external clients from making TCP connections with internal clients, but allows internal clients to connect to outside.

Stateless packet filtering: more examples

Where is a firewall implemented?

<table>
<thead>
<tr>
<th>Policy</th>
<th>Firewall Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>No outside Web access.</td>
<td>Drop all outgoing packets to any IP address, port 80</td>
</tr>
<tr>
<td>No incoming TCP connections, except</td>
<td>Drop all incoming TCP SYN packets to any IP except 130.207.244.203, port 80</td>
</tr>
<tr>
<td>those for institution's public Web server only</td>
<td></td>
</tr>
<tr>
<td>Prevent Web-radios from eating up the available bandwidth.</td>
<td>Drop all incoming UDP packets - except DNS and router broadcasts.</td>
</tr>
<tr>
<td>Prevent your network from being used for a DoS attack.</td>
<td>Drop all ICMP packets going to a &quot;broadcast&quot; address (eg 130.207.255.255).</td>
</tr>
<tr>
<td>Prevent your network from being tracerouted.</td>
<td>Drop all outgoing ICMP TTL expired traffic</td>
</tr>
</tbody>
</table>

Access Control Lists

**ACL:** table of rules, applied top to bottom to incoming packets: (action, condition) pairs

<table>
<thead>
<tr>
<th>action</th>
<th>source address</th>
<th>dest address</th>
<th>protocol</th>
<th>source port</th>
<th>dest port</th>
<th>flag bit</th>
</tr>
</thead>
<tbody>
<tr>
<td>allow</td>
<td>222.22/16</td>
<td>outside of 222.22/16</td>
<td>TCP</td>
<td>&gt; 1023</td>
<td>80</td>
<td>any</td>
</tr>
<tr>
<td>allow</td>
<td>outside of 222.22/16</td>
<td>222.22/16</td>
<td>TCP</td>
<td>80</td>
<td>&gt; 1023</td>
<td>ACK</td>
</tr>
<tr>
<td>allow</td>
<td>222.22/16</td>
<td>outside of 222.22/16</td>
<td>UDP</td>
<td>&gt; 1023</td>
<td>53 (DNS)</td>
<td>---</td>
</tr>
<tr>
<td>allow</td>
<td>outside of 222.22/16</td>
<td>222.22/16</td>
<td>UDP</td>
<td>53</td>
<td>&gt; 1023</td>
<td>---</td>
</tr>
<tr>
<td>deny</td>
<td>all</td>
<td>all</td>
<td>all</td>
<td>all</td>
<td>all</td>
<td>all</td>
</tr>
</tbody>
</table>
Stateful packet filtering

- Stateless packet filter: heavy handed tool
  - Admits packets that “make no sense,” e.g., dest port = 80, ACK bit set, even though no TCP connection established:

<table>
<thead>
<tr>
<th>action</th>
<th>source address</th>
<th>dest address</th>
<th>protocol</th>
<th>source port</th>
<th>dest port</th>
<th>flag bit</th>
</tr>
</thead>
<tbody>
<tr>
<td>allow</td>
<td>outside of 222.22/16</td>
<td>222.22/16</td>
<td>TCP</td>
<td>80</td>
<td>&gt;1023</td>
<td>ACK</td>
</tr>
</tbody>
</table>

- Stateful packet filter: track status of every TCP connection
  - Track connection setup (SYN), teardown (FIN): can determine whether incoming, outgoing packets “makes sense”
  - Timeout inactive connections at firewall: no longer admit packets

Application gateways

- Filter packets on application data as well as on IP/TCP/UDP fields.
  - Example: allow select internal users to telnet outside

  1. Require all telnet users to telnet through gateway.
  2. For authorized users, gateway sets up telnet connection to dest host.
     - Gateway relays data between 2 connections
  3. Router filter blocks all telnet connections not originating from gateway.

Limitations of firewalls

- IP spoofing: router can’t know if data “really” comes from claimed source
- Filters often use all or nothing policy for UDP.
- Tradeoff: degree of communication with outside world, level of security
- Many highly protected sites still suffer from attacks.

Intrusion detection systems

- Deep packet inspection: look at packet contents (e.g., check character strings in packet against database of known virus, attack strings)
- Examine correlation among multiple packets
  - port scanning
  - network mapping
  - DoS attack
Network Security (summary)

Basic techniques......
- cryptography (symmetric and public)
- message integrity
- end-point authentication

... used in many different security scenarios
- secure email
- secure transport (SSL)

Operational Security: firewalls and IDS