Securing Computer Networks

- Recall the five layers in computer networks:

  - Application
  - Transport
  - Network
  - Link
  - Physical

- Q: In what layer do you think we should apply security tool?
Securing Computer Networks

- Recall the five layers in computer networks:

  ![Network Layers Diagram]

  We secure everything coming from above.

- Q: In what layer do you think we should apply security tool?
Recall the five layers in computer networks:

We secure everything coming from above.

Q: In what layer do you think we should apply security tool?
Security in Application Layer

- We design a security module for a specific application:

- Different applications may need different security efforts.

Case Study:

- Securing Emails

- Q: What are the security needs for emails?
Secure e-mail: Confidentiality

- Alice wants to send confidential e-mail, m, to Bob.

Notations:
- $K_S$: A symmetric key
- $K_B$: Bob’s public key
- $K_S$: Bob’s private key
Secure e-mail: Confidentiality

Alice wants to send confidential e-mail, m, to Bob.

Alice:
- Generates a random symmetric key $K_S$
- Encrypts message with $K_S$
- Also encrypts $K_S$ with Bob's public key
- Sends both $K_S(m)$ and $K_B(K_S)$ to Bob
Secure e-mail: Confidentiality

- Alice wants to send confidential e-mail, m, to Bob.

\[ \text{K}_S(m) \]

\[ \text{K}_B(\text{K}_S) \]

\[ \text{K}_B^-(\text{m}) \]

\[ \text{K}_S^- \]

\[ \text{K}_B^+ \]

\[ \text{K}_S \]

Bob:

- Uses his private key to decrypt and recover \( \text{K}_S \)
- Uses \( \text{K}_S \) to decrypt \( \text{K}_S(m) \) to recover \( m \)
Secure e-mail: Authentication

- Alice wants to provide sender authentication message integrity

- Alice digitally signs message

- Sends both message and digital signature
Secure e-mail: Authentication

- Alice wants to provide both secrecy and authentication.

Alice uses three keys: her private key, Bob’s public key, newly created symmetric key
Secure e-mail: Authentication

Alice wants to provide both secrecy and authentication.

\[ H(\cdot), K_A^-(\cdot), K_A^-(H(m)), +, K_S(\cdot), +, K_B^+(\cdot), \]

Q: What are the steps on the Bob's side?
Secure e-mail: Example

- **PGP (Pretty Good Privacy) Protocol**

\[
\begin{align*}
\text{m} & \xleftarrow{} \text{BEGIN PGP SIGNED MESSAGE-----} \\
& \text{Hash: SHA1} \\
& \text{Bob:} \\
& \text{Can I see you tonight?} \\
& \text{Passionately yours, Alice} \\
& \text{-----BEGIN PGP SIGNATURE-----} \\
& \text{Version: PGP for Personal Privacy 5.0} \\
& \text{Charset: noconv} \\
& \text{yhHJRHHhGJGhg/12EpJ+lo8gE4vB3mqJhvFEmZP9t6n7G6m5Gw2} \\
& \text{-----END PGP SIGNATURE-----} \\
\end{align*}
\]
Secure e-mail: Example

- PGP (Pretty Good Privacy) Protocol

\[
\begin{align*}
\text{m} & \xrightarrow{\text{K}_A^- (H(m))} \\
\text{K}_B^+ (\ldots) & \xrightarrow{\text{-----BEGIN PGP MESSAGE-----}} \\
\text{Version: PGP for Personal Privacy 5.0} \\
\text{u2R4d+/jKmn8Bc5/+hgDsqAewsDfrGdszX68iikm5F6Gc4sDfcXyt} \\
\text{RfdSIojuHgbcfDssWe7/K=IkhnMikLo0+I/BvcX4t==Ujk9PbcD4} \\
\text{Thdf2awQgHbnmKlok8iy6gThlp} & \xrightarrow{\text{-----END PGP MESSAGE-----}} \\
\end{align*}
\]
Security in Transport Layer

• This is not application-specific.

Case Study:

- **SSL**: Secure Sockets Layer
- Transport Layer Security to TCP-based apps.
SSL: Secure Sockets Layer

• Widely deployed protocol
  – Supported by almost all browsers, web servers
  – HTTPS

• First Implementation:
  – Netscape

• Provides
  – Confidentiality
  – Data Integrity
  – Authentication

• Original goals:
  – Web e-commerce transactions
  – Encryption (especially credit-card numbers)
  – Web-server authentication
  – Optional client authentication

• Available to all TCP apps
  – Secure socket interface
SSL Example – Online Purchase

- **Confidentiality**: Credit Card Information
- **Data Integrity**: Altering Your order (e.g., address)
- **Server Authentication**: Fake Amazon Website!
**SSL and TCP/IP**

<table>
<thead>
<tr>
<th>Application</th>
<th>SSL</th>
<th>Application</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCP</td>
<td></td>
<td>TCP</td>
</tr>
<tr>
<td>IP</td>
<td></td>
<td>IP</td>
</tr>
</tbody>
</table>

normal application  

application with SSL

- SSL provides application programming interface (API) to application developers to use “secure sockets”.

- C and Java SSL libraries/classes readily available
SSL Approach: Something Like PGP

- But want to send byte streams & interactive data (Q: Why?)
- Want set of secret keys for entire connection
- Want certificate exchange as part of protocol: handshake phase
SSL Lifetime

• **Handshake**: Alice and Bob use their certificates, private keys to authenticate each other.
  
  – Now, they can exchange shared secret

• **Key Derivation**: Alice and Bob use shared secret to derive set of [symmetric] keys.

• **Data Transfer**: Exchange encrypted byte streams.

• **Connection Closure**: special messages to securely close connection (Q: Why securely?)
SSL: A Simple Handshake

**MS:** Master Secret  ➔  To Generate Symmetric Keys

**EMS:** Encrypted Master Secret

[ More details on handshake coming soon! ]
SSL: Key Derivation

• Generate and use different keys for
  – Message Authentication Code (MAC)
  – Data Encryption

• Using Key Derivation Function (KDF) and Master Key (MS), Bob and Alice generate four different keys:
  – $K_B = \text{Encryption key for data sent from Bob to Alice}$
  – $M_B = \text{MAC key for data sent from Bob to Alice}$
  – $K_A = \text{Encryption key for data sent from Alice to Bob}$
  – $M_A = \text{MAC key for data sent from Alice to Bob}$
SSL: Data Records

• Break down the byte stream into series of records
  – Each record carries a separate MAC
  – Receiver can act on each record as it arrives
    • E.g., separate message identification in instant messaging

• Issue: Receiver needs to distinguish MAC from data
  – Want to use variable-length records → Indicate Data Length
SSL: Sequence Numbers

• *Problem*: attacker can capture and replay record or re-order records – *Is it a new message?*
SSL: Sequence Numbers

• **Problem:** attacker can capture and replay record or re-order records – **Is it a new message?**

• **Solution:** put sequence number into MAC:
  – MAC = MAC($M_x$, sequence | | data)
  – Note: no separate sequence number field (Unlike TCP)
SSL: Sequence Numbers

• **Problem:** attacker can capture and replay record or re-order records – *Is it a new message?*

• **Solution:** put sequence number into MAC:
  – $\text{MAC} = \text{MAC}(M_x, \text{sequence} \mid \mid \text{data})$
  – Note: *no separate* sequence number field (Unlike TCP)

• **Problem:** attacker could replay all records
• **Solution:** use nonce (a number used once in a life-time)
SSL: Control Information

• **Problem:** truncation attack:
  – Attacker forges TCP connection close segment
  – One or both sides think there is less data than actually is.

• **Solution:** record types, with one type for closure
  – Type 0 for data; Type 1 for closure

• MAC = MAC(M_x, sequence || type || data)
SSL: Summary

- hello
- certificate, nonce
- $K_B^+(MS) = EMS$
- type 0, seq 1, data
- type 0, seq 2, data
- type 0, seq 1, data
- encrypted
- type 0, seq 3, data
- type 1, seq 4, close
- type 1, seq 2, close

bob.com
SSL: We Are Not Done Yet!

• How long are fields? → Standard

• Which encryption protocols?

• Want negotiation?
  – Allow Bob (client) and Alice (server) to support different encryption algorithms
  – Allow Bob (client) and Alice (server) to choose together specific algorithm before data transfer.
SSL Encryption Protocol Selection

• Cipher Suite
  – Public-key algorithm
  – Symmetric encryption algorithm
  – MAC algorithm

• SSL supports many cipher suites
  – DES – Data Encryption Standard: block
  – 3DES – Triple strength: block
  – RC2 – Rivest Cipher 2: block
  – RC4 – Rivest Cipher 4: stream
SSL: Handshake Details

**Purpose**

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

1. Client sends list of algorithms it supports
   - Along with client nonce

2. Server chooses algorithms from list; sends back:
   choice + public key certificate + server nonce
SSL: Handshake Details

3. Client

- Verifies server’s public key certificate with a CA
- Generates master_secret,
- Encrypts it using server’s public key
- Sends the encryption result back to the server

• Using its private key, the server obtains master_secret.
SSL: Handshake Details

4. Client and server use
   – master_secret
   – Nonces

   to independently compute:
   – Encryption
   – MAC keys
SSL: Handshake Details

5. Client sends a MAC in all the handshake messages

6. Server sends a MAC in all the handshake messages
SSL Programming

- SSL programming tutorial:
  http://h71000.www7.hp.com/doc/83final/ba554_90007/ch04s03.html
Security in Network Layer

• This is not application-specific.

We secure everything coming from above.

• Case Study:

❖ **IPSec**: Internet Protocol (IP) Security Protocol

❖ **VPN**: Virtual Private Networks (Using IPSec)
What is Network-layer Confidentiality?

*Between two network entities:*

- Sending entity encrypts **datagram** (Q: why?) payload.

- Payload could be:
  - TCP or UDP segment, ICMP message, ...

- Indifferent to the content:
  - Web pages, e-mail, P2P file transfers, TCP SYN packets ...

- “Blanket Coverage”
Motivation:

• Institutions often want private networks for security.
  – Cost: Separate routers, links, DNS infrastructure.
  – Feasibility: Members are not always physically co-located.

• VPN: Inter-office traffic is sent over public Internet, but
  – Encrypted before entering public Internet
  – Logically separated from other traffic
Virtual Private Networks (VPNs)

Router w/ IPv4 and IPsec

Router w/ IPv4 and IPsec

Headquarters

Branch Office

Salesperson in a Hotel

Laptop w/ IPsec

IP header
IPsec header
Secure payload

IP header
IPsec header
Secure payload

public Internet

Laptop w/ IPsec
Virtual Private Networks (VPNs)

- **Headquarters**
  - Router w/ IPv4 and IPsec
  - IPs

- **Branch Office**
  - Router w/ IPv4 and IPsec
  - Secure payload

- **Laptop in a Hotel**
  - Laptop w/ IPsec

- **Salesperson in a Hotel**
  - No Need for IPsec
Virtual Private Networks (VPNs)

- **Public Internet**
- **Router w/ IPv4 and IPsec**
  - **IP header**
  - **IPsec header**
  - **Secure payload**
- **Laptop w/ IPsec**
- **Salesperson in a Hotel**
- **Must Use IPsec**

Diagram showing network connectivity with emphasis on the use of IPsec for secure communication.
IPsec Services

• Data integrity
• Origin authentication
• Replay attack prevention (Sliding Window)
• Confidentiality

• Two protocols to send and receive secure datagrams:
  – Authentication Header (AH)
  – Encapsulated Security Payload (ESP)
IPsec Services

• Authentication Header (AH)
  – Source Authentication
  – Data Integrity

• Encapsulated Security Payload (ESP)
  – Source Authentication
  – Data Integrity
  – Confidentiality
IPsec Services

• Authentication Header (AH)
  – Source Authentication
  – Data Integrity

• Encapsulated Security Payload (ESP)
  – Source Authentication
  – Data Integrity
  – Confidentiality Critical
IPsec Services

• Authentication Header (AH)
  — Source Authentication
  — Data Integrity

• Encapsulated Security Payload (ESP)
  — Source Authentication
  — Data Integrity
  — Confidentiality

More Popular
(Our Focus)
IPsec – Tunneling Mode

- Edge routers IPsec-aware

- Hosts IPsec-aware
Security Associations (SAs)

- In ESP, before sending data, a “security association (SA)” is established from sending to receiving entity.

- Receiving entities maintain *state information* about SA:
  - SA is opened
  - SA is closed

- **Details**: How many SAs in VPN w/ headquarters, branch office, and n traveling salespeople?
Example SA from R1 to R2

`security association`
Example SA from R1 to R2

initiating router

security association

172.16.1/24

200.168.1.100

193.68.2.23

172.16.2/24

headquarters

Internet

branch office

R1

R2

network security, part 3, UCR & TTU

summer workshop on cyber security August 12-16, 2013
Example SA from R1 to R2

172.16.1/24

200.168.1.100

Internet

193.68.2.23

R1

security association

R2

172.16.2/24

headquarters

branch office

Receiving Router
Example SA from R1 to R2

R1 maintains the following state information:

- 32-bit SA identifier: Security Parameter Index (SPI)
- Origin SA interface (200.168.1.100)
- Destination SA interface (193.68.2.23)
- Encryption key and type of encryption (confidentiality)
- Authentication key and type of authentication
Example SA from R1 to R2

**Example SA Fields:**

- SPI: 12345
- Source IP: 200.168.1.100
- Dest IP: 193.68.2.23
- Protocol: ESP
- Encryption algorithm: 3DES-cbc
- HMAC algorithm: MD5
- Encryption key: 0x7aeaca...
- HMAC key: 0xc0291f...
Security Association Database (SAD)

- Routers hold SA states in *security association database (SAD)*, where it can locate them during processing.

- When sending IPsec datagram, R1 uses SAD to see how to process datagram. *(Q: what kind of process?)*

- When IPsec datagram arrives to R2, R2 indexes SAD with SPI, and processes datagram accordingly.
Focus for now on tunnel mode with ESP

**IPsec Datagram**

- **encrypted**
- **authenticated**

Diagram:
- New IP header
- ESP hdr
- Original IP hdr
- Original IP datagram payload
- ESP trl
- ESP MAC
- SPI
- Seq #
- Padding
- Pad length
- Next header
IPsec Datagram

Focus for now on tunnel mode with ESP

---

**new IP header**  |  **ESP hdr**  |  **original IP hdr**  |  **Original IP datagram payload**  |  **ESP trl**  |  **ESP MAC**

- **SPI**
- **Seq #**
- **padding**
- **pad length**
- **next header**

**authenticated**  \(\xrightarrow{\text{encrypted}}\)

---

Adjust Length for Block Ciphers
IPsec Datagram

Focus for now on tunnel mode with ESP

 authenticated

 encrypted

new IP header | ESP hdr | original IP hdr | Original IP datagram payload | ESP trl | ESP MAC

SPI | Seq # | padding | pad length | next header

UDP, TCP, etc.
Focus for now on tunnel mode with ESP

Authentication with Shared Secret Key

```
<table>
<thead>
<tr>
<th>new IP header</th>
<th>ESP hdr</th>
<th>original IP hdr</th>
<th>Original IP datagram payload</th>
<th>ESP trl</th>
<th>ESP MAC</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPI</td>
<td>Seq #</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>padding</td>
<td>pad length</td>
<td>next header</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

- authenticated
- encrypted
IPsec Big Picture: What happens?

172.16.1/24

172.16.2/24

headquarters

branch office

R1

Internet

R2

200.168.1.100

193.68.2.23

security association

new IP header

ESP hdr

original IP hdr

Original IP datagram payload

ESP trl

ESP MAC

SPI

Seq #

padding

pad length

next header

authenticated

encrypted
IPsec Sequence Numbers

• For new SA, sender initializes seq. # to 0

• Each time datagram is sent on SA:
  – sender increments seq # counter and updates seq # field

Q: Why do we use sequence numbers here?
IPsec Sequence Numbers

• For new SA, sender initializes seq. # to 0

• Each time datagram is sent on SA:
  – sender increments seq # counter and updates seq # field

• **Goal:**
  – Prevent attacker from sniffing and **replaying** a packet
IPsec Sequence Numbers

• For new SA, sender initializes seq. # to 0

• Each time datagram is sent on SA:
  – sender increments seq # counter and updates seq # field

• **Goal:**
  – Prevent attacker from sniffing and **replaying** a packet

• **Method:**
  – destination checks for **duplicates**
  – doesn’t keep track of *all* received packets; just uses a **window**
IPsec Services (Summary)

- Trudy sits between R1 & R2. With no keys, can she:
  - See the original contents of datagram?
  - See the source, dest IP addr, transport protocol, app port?
  - Flip bits without detection?
  - Pretend to be R1 using R1’s IP address?
  - Replay a datagram?

Justify Your Answers!
IKE: Internet Key Exchange

• For 1 or 2 tunnels we can do manual keying.
  – Just manually enter the SA information to two routers
    • The router in Headquarters.
    • The router in Branch Office.

• It will be impractical for VPN with 100s of endpoints!

• Instead use **IPsec IKE (Internet Key Exchange)**
  – Similar to the Handshaking in SSL (Q: Remember SSL?)
IKE: PSK and PKI

- Authentication (prove who you are) with either
  - Pre-shared secret (PSK) or
  - With PKI (public/private keys and certificates).
IKE: PSK and PKI

• Authentication (prove who you are) with either
  – Pre-shared secret (PSK) or
  – With PKI (public/private keys and certificates).

• **PSK**: both sides start with a secret
  – run IKE to authenticate each other and to generate IPsec SAs (one in each direction) with encryption, authentication keys
IKE: PSK and PKI

• Authentication (prove who you are) with either
  – Pre-shared secret (PSK) or
  – With PKI (public/private keys and certificates).

• **PSK**: both sides start with a secret
  – run IKE to authenticate each other and to generate IPsec SAs
    (one in each direction) with encryption, authentication keys

• **PKI**: both sides start with public/private keys, certificate
  – run IKE to authenticate each other, obtain IPsec SAs
IPsec Summary

• IKE message exchange for algorithms, keys, SPI #s

• ESP Protocol (or AH protocol)
  – Use SAD, Sequence #s, etc.

• IPsec peers can be two end systems:
  – Two routers/firewalls,
  – A router/firewall and an end system
  – Two end systems, ...
Security in Link Layer

• This is **not** application-specific.

• **Case Study:**

  ❖ **Wireless LAN:** Wireless Local Area Network
  
  ❖ **WEP:** Wired Equivalent Privacy
Security in Link Layer

• The main focus at link layer is to handle different link types:
  – Wired (Ethernet Cable, etc.)
  – Wireless (WLAN: Shared Channels, Q: Why?)

• Q: Which one is more vulnerable? Why?

• Q: Why do we care about WEP (Wired Equivalent Privacy)?
Security in Link Layer

• Our Focus:

WiFi = IEEE 802.11

• We will cover:

  – IEEE 802.11 WEP

  – IEEE 802.11i WEP
WEP Mobile Device Authentication

• WEP does **not** specify a key management algorithm.

• Assumption:
  
  – Shared key via an **out-of-band** method.

(Q: What does that mean?)
WEP Mobile Device Authentication

• WEP does not specify a key management algorithm.
WEP Mobile Device Authentication

• WEP does **not** specify a key management algorithm.

• **Q:** Should the Mobile Device send the Security Key to the Access Point to Authenticate Itself? **Q:** Why?
WEP Mobile Device Authentication

- WEP does **not** specify a key management algorithm.

- **Q:** Should the Mobile Device send the Security Key to the Access Point to Authenticate Itself? **Q:** Why?

- **Q:** What is your suggestion?
WEP Mobile Device Authentication

• **Step 1**: Mobile Device Requests Authentication.

![Diagram showing AP and Mobile Device connected with 'I need access' message.]
### WEP Mobile Device Authentication

- **Step 2**: Access point responds with a 128 byte nonce.

![Diagram showing WEP Mobile Device Authentication process](image)
WEP Mobile Device Authentication

• **Step 3**: MD encrypts nonce using its Security Key.
WEP Mobile Device Authentication

- **Step 4**: AP checks encrypted nonce and grants access.
WEP Mobile Device Authentication

- **Step 4**: AP checks encrypted nonce and grants access.

[ AP indicates if Authentication is *necessary* in beacon frames. ]
WEP Mobile Device Authentication

• A hacker may **hear** all message exchanges. **Q: Why?**
WEP Mobile Device Authentication

- Q: Does it help to try replay? Q: Why?
WEP Mobile Device Authentication

• Q: How about trying different security keys... 😊
Security Issues in Wireless Link Layer

• **Note 1:** WEP must be “self-synchronized”
  
  – Each frame should be *separately* encrypted.
  
  – That is, no encryption dependency among frames.
  
  – **Q:** Why?
Security Issues in Wireless Link Layer

• **Note 1:** WEP must be “self-synchronized”
  
  – Each frame should be *separately* encrypted.
  
  – That is, *no encryption dependency* among frames.
  
  – **Q:** Why?

  Frame _Loss_ is a Frequent Problem in Wireless Links.
Security Issues in Wireless Link Layer

• **Note 2:** We need to handle bit errors with CRC:

  ─ CRC: Cyclic Redundancy Check

  ─ Include CRC inside encryption.
Security Issues in Wireless Link Layer

• **Note 3**: Minimal computation for encryption:
  
  – We cannot make encryption the **bottleneck**.
  
  – We send **thousands of frames** in a second.

  • Can’t afford spending too much time on encryption.
  
  • CRC an other tasks already take some time.
Security Issues in Wireless Link Layer

- IEEE 802.11 WEP uses **XOR** encryption!

![Diagram of XOR encryption process]

- **Q**: What is the benefit?
Security Issues in Wireless Link Layer

- IEEE 802.11 WEP uses **XOR** encryption!

![Diagram showing the process of an XOR encryption process.](image-url)

- **Q:** What is the disadvantage?
Q: What if the hacker knows the content?

- Example: She knows what file is being downloaded.
- Knowing the content of one frame is enough:
Security Issues in Wireless Link Layer

• The hacker can apply the key to the rest of frame.

• Solution: Use a different key for each frame.
  – Of course, both sides should know the key.
  – Q: Any suggestion?
Security Issues in Wireless Link Layer

- IEEE 802.11 WEP uses **Initialization Vector (IV):**

  Randomly Generated
  24-bit IV
  40-bit Security Key
  Key Generator Function
Security Issues in Wireless Link Layer

- IEEE 802.11 WEP uses **Initialization Vector (IV):**

  - Randomly Generated 24-bit IV
  - 40-bit Security Key

  Key Generator Function

  Encrypt Frame Data

  Indicate IV (Q: Why?)

  Per-Frame Security Key
Security Issues in Wireless Link Layer

- IEEE 802.11 WEP uses **Initialization Vector (IV)**:

```
+-----------------+---------+-----------------+--------+
| 802.11 Header  | IV      | Original Payload| CRC    |
|                 |         |                 |        |
```

Encrypted Link Layer Payload
Security Issues in Wireless Link Layer

- IEEE 802.11 WEP uses **Initialization Vector (IV):**

```
+-------------------------------+----------+-------------------+----------+
| 802.11 Header                | IV       | Original Payload  | CRC      |
+-------------------------------+----------+-------------------+----------+
```

- Encrypted

- Link Layer Payload

- **Q:** Do we encrypt IV or is it in plain text?
Security Issues in Wireless Link Layer

- IEEE 802.11 WEP uses **Initialization Vector (IV)**:

  - **Q**: Do we encrypt IV or is it in plain text?
  - **Q**: Should we avoid duplicating ‘random’ IVs?
IEEE 802.11i: Improved Security

• Numerous (stronger) forms of encryption:
  – Encryption method is chosen during handshake.

• Uses authentication server separate from AP
  – MDs use AP to talk to authentication server first.
  – Example: Institutional WiFis, like at UC Riverside.

• Provides key distribution and generation, etc…
Firewalls: Operational Security

**Firewall**

Isolates organization’s internal net from larger Internet, allowing some packets to pass, blocking others!
Why Do We Need Firewalls?

• **Reason 1:** They can Prevent **Denial-of-Service** (DoS) Attacks:
  
  - TCP SYN Flooding:
    
    - Attacker establishes many bogus TCP connections, no resources left for “real” connections

• **Reason 2:** Prevent illegal modification / access of internal data

• **Q:** Other reasons?
Firewall Example: Packet Filtering

- Internal network connected to Internet via *router firewall*.
Firewall Example: Packet Filtering

Router *filters packet-by-packet* based on:

- Source IP address, Destination IP address

*Example:* Block TCP connections by IP addresses in a county.

Should arriving packet be allowed in? Departing packet let out?
Router filters packet-by-packet based on:

- TCP/UDP source and destination port numbers

  • Example: Block all UDP packets, Or block non-standard ports
Firewall Example: Packet Filtering

• Router *filters packet-by-packet* based on:
  
  – ICMP message type

  • **Example:** Block any “Ping” from the outside.
Firewall Example: Packet Filtering

• Router *filters packet-by-packet* based on:

  – TCP SYN and ACK bits

• **Example:** Block any “inbound” ACK = 0. Only the internal hosts can “initiate” a TCP connection to outside; not the reverse.
Netowrk Security (Summary)

Basic Techniques......

– Cryptography (Symmetric and Public)
– Message Integrity
– End-point Authentication

.... used in many different layers

– Application Layer: Secure Email
– Transport Layer: Secure Transport (SSL)
– Network Layer: IP sec
– Link Layer: 802.11
References

General Overview:

More Specific Network Security Topics: