2.2. Threats in Networks (1)

- Outline
  - a) Introduction
  - b) Network vulnerabilities
  - c) Who attacks networks?
  - d) Threat precursors
  - e) Threats in transit: eavesdropping and wiretapping
  - f) Protocol flaws
  - g) Types of attacks:
    - g-1) Impersonation
    - g-2) Spoofing
    - g-3) Message confidentiality threats
    - g-4) Message integrity threats
    - g-5) Web site attacks

Threats in Networks (2)

- Outline—cont.

  g) Types of attacks-cont.:

  - g-6) Denial of service
  - g-7) Distributed denial of service
  - g-8) Threats to active or mobile code
  - g-9) Scripted and complex attacks

  h) Summary of network vulnerabilities

a. Introduction (1)

- We will consider
  threats aimed to compromise C-I-A
  applied against data, software, or hardware
  by nature, accidents, nonmalicious humans, or malicious attackers

Introduction (2)

- From CSI/FBI Report 2002 (survey of ~500 com/gov/edu/org)
  - 90% detected computer security breaches
  - 80% acknowledged financial losses
  - 44% (223) were willing/able to quantify losses: $455M
  - Most serious losses: theft of proprietary info and fraud
    - 26 respondents: $170M
    - 25 respondents: $115M
  - 74% cited *Internet connection* as a frequent point of attack
  - 33% cited *internal systems* as a frequent point of attack
  - 34% *reported* intrusions to law enforcement (up from 16%-1996)

[cf. D. Frincke]

Introduction (3)

- More from CSI/FBI Report 2002
  - 40% detected external penetration
  - 40% detected DoS attacks
  - 78% detected employee abuse of Internet
  - 85% detected computer viruses
  - 38% suffered unauthorized access on Web sites
  - 21% didn’t know
  - 12% reported theft of information
  - 6% *reported* financial fraud (up from 3%-- 2000)

[cf.: D. Frincke]

b. Network vulnerabilities (1)

- Network characteristics significantly increase security risks
- These vulnerability-causing characteristics include:

1) Attacker anonymity
    - Attacker can be far away
    - Can disguise attack origin (pass through long chain of hosts)
        - Weak link: computer-to-computer authentication

2) Many points of origin and target for attacks
    - Data and interactions pass through many systems on their way between user and her server
    - Each system can be origin of an attack or target for attack
        - Systems might have widely different security policies/mechanisms

Network vulnerabilities (2)

3) Resource and workload sharing
   - More *users* have access to networks than to stand-alone systems
   - More *systems* have access to networks

4) Network complexity
   - Complexity much higher in networks than in single OSs

5) Unknown or dynamic network perimeter
   - Dynamic in any network, unknown in network w/o single administrative control
     - Any new host can be untrustworthy
   - Administrator might not known that some of hosts of *his* network are also hosts in *another* network
     - Hosts are free to join other networks

Network vulnerabilities (3)

6) Uknown paths between hosts and users
   - Many paths
   - Network decides which one chosen
     - Network might change path any time

7) Nonuniform security policies/mechanisms for hosts belonging to multiple networks
   - If Host H belongs to N1 and N2, does it follow:
     - N1's rules?
     - N2's rules?
     - Both?
       - What if they conflict?

c. Who attacks networks? (1)

- Who *are* the attackers?
    - We don’t have a name list
- Who the attackers *might be*?
    - MOM will help to answer this
        - MOM = Method/Opportunity/Motive
- Motives of attackers:

1) Challenge/Power
2) Fame
3) Money/Espionage
4) Ideology

Who attacks networks? (2)

1) Attacking for challenge/power
   - Some enjoy intellectual challenge of defeating supposedly undefeatable
   - Successful attacks give them sense of power
   - Not much challenge for vast majority of hackers
     - Just replay well-known attacks using scripts

2) Attacking for fame
   - Some not satisfied with challenge only
   - Want recognition – even if by pseudonym only
     - Thrilled to see their pseudonym in media

3) Attacking for money/espionage
   - Attacking for direct financial gains
   - Attacking to improve competitiveness of ones com/org
     - 7/2002: Princeton admissions officers broke into Yale's system
   - Attacking to improve competitiveness of ones country
     - Some countries support *industrial espionage* to aid their own industries

(cont.)

Who attacks networks? (3)

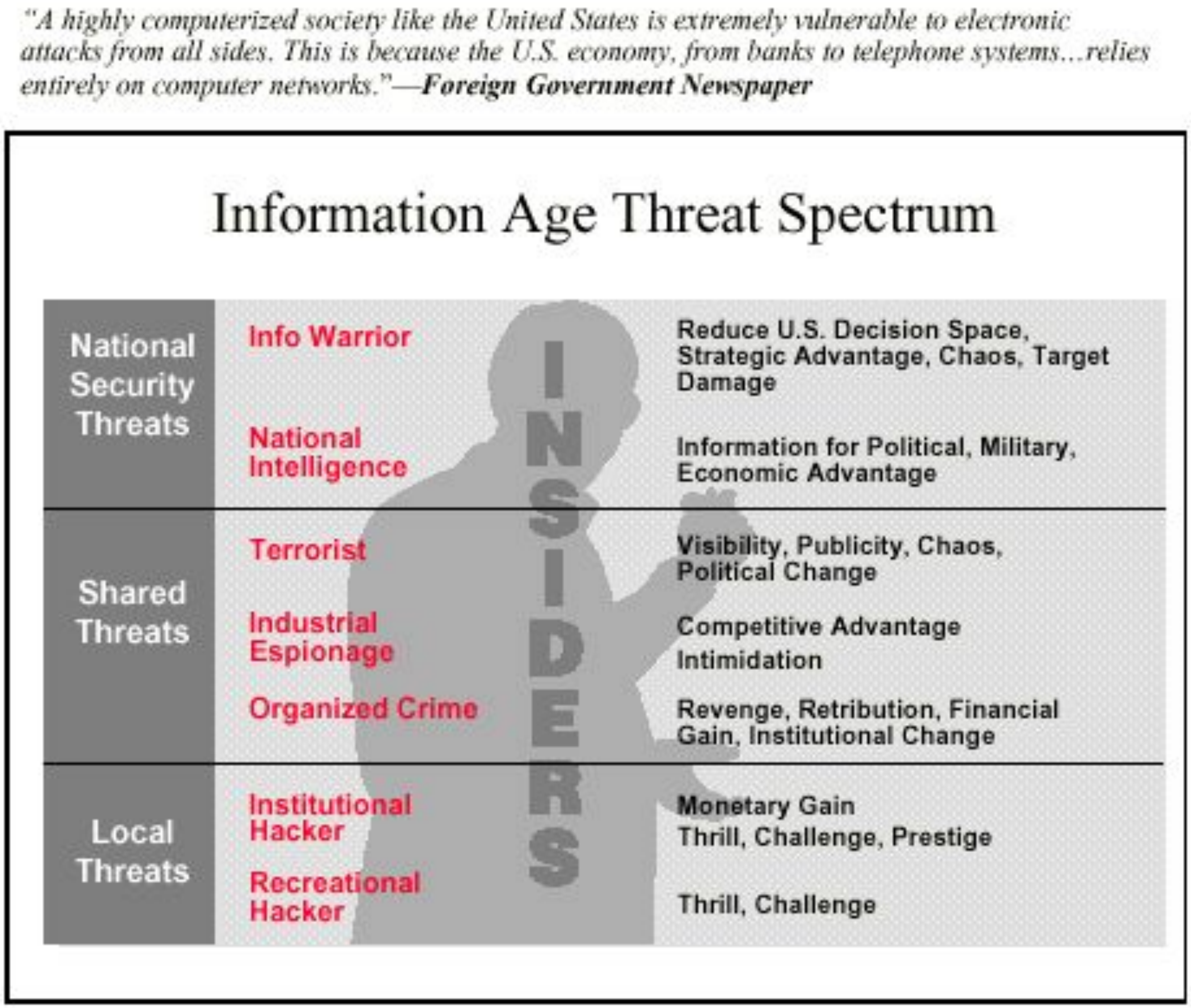
- Attacking to spy on/harm another country
  - Espionage and information warfare
    - Steal secrets, harm defense infrastructure, etc.
- Few reliable statistics – mostly *perceptions* of attacks
  - 1997-2002 surveys of com/gov/edu/org: ~500 responses/yr
    - 38-53% believed they were attacked by US competitor
    - 23-32% believed they were attacked by foreign competitor

5) Attacking to promote ideology

- Two types of ideological attacks:
  - Hactivism
    - Disrupting normal operation w/o causing serious damage
  - Cyberterrorism
    - Intent to seriously harm
      - Including loss of life, serious economic damage

Who attacks networks? (4)

Recall: Threat Spectrum



"A highly computerized society like the United States is extremely vulnerable to electronic attacks from all sides. This is because the U.S. economy, from banks to telephone systems…relies entirely on computer networks."—***Foreign Government Newspaper***

Information Age Threat Spectrum

INSIDERS

National Security Threats	Info Warrior	Reduce U.S. Decision Space, Strategic Advantage, Chaos, Target Damage
	National Intelligence	Information for Political, Military, Economic Advantage
Shared Threats	Terrorist	Visibility, Publicity, Chaos, Political Change
	Industrial Espionage	Competitive Advantage Intimidation
	Organized Crime	Revenge, Retribution, Financial Gain, Institutional Change
Local Threats	Institutional Hacker	Monetary Gain Thrill, Challenge, Prestige
	Recreational Hacker	Thrill, Challenge

[cf. D. Frincke]

Who attacks networks? (5)

- What about *moral objections* to harming others?
  - Some believe they'll cause no harm
  - Some believe that demonstrating system weakness serves public interest (even if there's some harm)
  - Some don't have any moral objections

- They are all wrong!!!
  - There is no harmless attack
    - Harm can be as small as just using targets processor cycles
  - Any mistake can change a harmless attack into a very harmful attack
    - E.g., The Internet (Morris) Worm (1988)

d. Threat precursors (1)

- How attackers prepare for attacks?
  - Investigate and plan

  These are *threat prescursors*

- If we detect threat precursors, we might be able to block attacks before they're launched

- Threat prescursors techniques include:
  1) Port scan
  2) Social engineering
  3) Reconnaissance
  4) OS and application fingerprinting
  5) Using bulletin boards and chats
  6) Getting available documentation

Threat precursors (2)

1) Port scan

Port scanner - pgm that scans port indicated by IP address

- Reports about:
  a) Standard ports/services running and responding
     - Recall (ex.): port 80–HTTP, 25-SMTP(e-mail), 23-Telnet

  b) OS installed on target system

  c) Apps and app versions on target system

  => Can infer which known vulnerabilities present
- Example: **`nmap`**
  - **`nmap –sP 192.168.100.*`**
    - Performs quick (20-30 s) ping scan („P”)
    - Notice wild card!
  - **`nmap –sT 192.168.100.102`**
    - Performs much slower (~10 min.) TCP port scan („T”)
  - OPTIONAL: more on nmap „Computer Security Lab Manual” (p.199)

Threat precursors (3)

1) Port scan – cont.

- Other port scanning tools:
  - `netcat` (free)
  - Many commercial port scanners:
    - Nessus (Nessus Corp.)
    - CyberCop Scanner (Network Associates)
    - Secure Scanner (Cisco)
    - Internet Scanner (Internet Security systems)
    - ...

Threat precursors (4)

2) Social engineering

= using social skills and personal interaction to get someone to reveal security-releveant info or do sth that permits an attack

- Impersonates sb inside an organization
  - Person in a high position (works best – by intimidation), co-worker, ...
- Often exploits sense of urgency
  - „My laptop has been stolen and I have an important presentation. Can you help me ....”
- Relies on human tendency to help others when asked politely

Threat precursors (5)

2) Social engineering – cont.

- Example: Phone call asking for system info
  - Never provide system info to a caller
  - Ask for identification
  - Best: Refer to help desk or proper system/security authority
  - If contact with sys/sec auth impossible, you might consider calling back but using phone number known to you from *independent source* (*not* the number given by the caller)
    - Independent source: known beforehand, obtained from company directory, etc.

3) Reconnaissance

= collecting discrete bits of security information from various sources and putting them together

- Reconnaissance techniques include:
  a) Dumpster diving
  b) Eavesdropping
     - E.g., follow employees to lunch, listen in
  c) Befriending key personnel (social engg!)

- Reconnaissance requires little training, minimal investment, limited time

  BUT can give big payoff in gaining background info

Threat precursors (7)

4) OS and application fingerprinting

= finding out OS/app name, manufacturer and version by using pecularities in OS/app responses

- Example: Attacker's approach
  - Earlier port scan (e.g., nmap) reveals that port 80 – HTTP is running
  - Attacker uses Telnet to send meaningless msg to port 80
  - Attacker uses response (or a lackof it) to infer which of many possible OS/app it is
    - Each version of OS/app has its fingerprint (pecularities) that reveals its identity (manufacturer, name, version)

Threat precursors (8)

5) Using bulletin boards / chats

- Attackers use them to help each other
  - Exchange info on their exploits, tricks, etc.

6) Getting available documentation

- Vendor documentation can help attackers
  - Esp. 3rd party developer documentation

e. Threats in transit: eavesdropping and wiretapping (1)

- Threats to data in transit:

1) Eavesdropping

   = overhearing *without any extra effort*

   E.g., admin anyway uses s/w to monitor network traffic to manage the network - in this way she effortlessly eavesdrops on the traffic

2) Wiretapping

   = overhearing *with some extra effort*

   a) Passive wiretapping

      Pretty similar to eavesdropping but some extra effort

      E.g., starting monitoring s/w usually not used

   b) Active wiretapping – injecting msgs

- Wiretapping technique depends on the communication medium

Threats in transit: eavesdropping and wiretapping (2)

- Wiretapping technique depends on the communication medium

1) Wiretapping cables

- Via *packet sniffer* for Ethernet or other LAN
  - Msgs broadcast onto Ethernet or other LAN
  - Reads all data packets—not only ones addressed to *this* node
- By means of *inductance*
  - Using radiation emitted by cable
  - Tap must be close to cable
- By *splicing* / connecting to cable
  - Can be detected by resistance/impedance change
- Note: If signal multiplexed (on WANs), wiretapper must extract packets of interest from intercepted data

Threats in transit: eavesdropping and wiretapping (3)

2) Wiretapping microwave

- Signal broadcast thru air, dispersed (cf. Fig. 7-14)
  => accessible to attackers
- Very insecure medium
- Protected by volume —carries a lot of various data, multiplexed

3) Wiretapping satellite links

- Very wide signal dispersion (even k*100 by n*1,000 mi)
  => easy to intercept
- Protected by being highly multiplexed

Threats in transit: eavesdropping and wiretapping (4)

4) Wiretapping optical fiber

- Must be tuned after each new connection made => easy to detect wiretaps (wiretaps destroy „balance”)
- Inductive tap impossible (no magnetic radiation for light)
- Easiest to tap at:
  - Repeaters, splices, and taps along the cable
  - Points of connection to computing equipment

5) Tapping wireless

- Typical signal range= interception range: 100-200 ft.
- Wireless communication standards:
  - 802.11b (≤10 Mbps)
  - 802.11a (~ 50 Mbps)
  - 802.11g – most popular currently
  - 802.11n – planned approval: Sept. 2007

cont.

Threats in transit: eavesdropping and wiretapping (5)

- Problem 1: Interception
  - Due to *no* encryption or *weak* encryption standard
  - 85% wireless installations don't provide encryption (!)
  - Standard encryption (WEP) is weak
    - WEP = Wired Equivalent Privacy
    - Stream cipher with 40- or 104-bit key
    - 40-bit key can be broken pretty easily
  - WEP superceded by:
    - WPA (Wi-Fi Protected Access) in 2003
    - Full IEEE 802.11i standard (also known as WPA2) in 2004
- Problem 2: Service theft
  - Popular DHCP protocol (negotiating with client) assigns one-time IP address *without authentication* (of the client)
    - DHCP = Dynamic Host Configuration Protocol
  - Anybody can get free Internet access (after she gets IP)

f. Protocol flaws

- Protocol flaws:
  - Design flaws
    - Proposed Internet protocols posted for public scrutiny
    - Does not prevent protocol design flaws
  - Implementation flaws

g. Types of attacks
g-1. Impersonation (1)

- *Impersonation* = attacker foils authentication and assumes identity of a *valid entity* in a communication
- Impersonation attack may be easier than wiretapping
- Types of impersonation attacks (IA):
  1) IA by guessing
  2) IA by eavesdropping/wiretaping
  3) IA by circumventing authentication
  4) IA by using lack of authentication
  5) IA by exploiting well-known authentication
  6) IA by exploiting trusted authentication

Impersonation (2)

1) Impersonation attacks by guessing

- Ways of guessing:
  - Common word/dictionary attacks
  - Guessing default ID-password pairs
    - E.g., GUEST-guest / GUEST-null / ADMIN-password
  - Guessing weak passwords
- Guessing can be helped by social engg
  - E.g., guess which account might be dead/dormant
    - Read in a college newspaper online that Prof. Ramamoorthy is on sabbatical => guessses that his acct is dormant
  - Social engg: call to help desk to reset password to one given by attacker

Impersonation (3)

2) Impersonation attacks by eavesdropping/wiretaping

- User-to-host or host-to-host authentication must not transmit password in the clear
  - Instead, e.g., transfer hash of a password
  - Correct protocols needed
    - Devil is in the details
  - Example of simple error: Microsoft LAN Manager
    - 14-char password of 67 characters
    - Divided into 2 pieces of 7 chars for transmission
    - Each piece hashed separately
    - To break hash, wiretapper need at most:

      $67^7 + 67^7 = 2 * 67^7$ attempts

      (as now each 7-char piece can be guessed separately)
    - Should have divided into 2 pieces for transmission *after* hashing, not before (hash 14 not 2 * 7 chrs) => would have 67^{14} possibilities (10 billion times more!)

Impersonation (4)

3) Impersonation attacks by circumventing authentication

- Weak/flawed authentication allows bypassing it
- „Classic" OS flaw:
  - Buffer overflow caused bypassing password comparison
  - Considered it correct authentication!
- Crackers routinely scan networks for OSs with weak/flawed authentication
  - Share this knowledge with each other

Impersonation (5)

4) Impersonation attacks by using lack of authentication

a) Lack of authorization by design

- Example: Unix facilitates host-to-host connection by users already authorized on their primary host
  - .rhosts - list of trusted hosts
  - .rlogin - list of trusted users allowed access w/o authentication
  - Attacker who gained proper id I1 on one host H1, can access all hosts that trust H1 (have H1 and I1 in .rhosts and .rlogin, respectively)

b) Lack of authorization due to administrative decision

- E.g., a bank may give access to public information to anybody under guest-no login account-pasword pair
- „Guest" account can be a foothold for attacker
  - Attacker will try to expand guest privileges to exploit the system

Impersonation (6)

5) Impersonation attacks by exploiting well-known authentic.

- Example: A computer manufacturer planned to use same login-password pair for maintenance account for any of its computers all over the world
- System/network admins often leave default password unchanged
  - Example: „community string" deafult password in SNMP protocol (for remote mgmt of network devices)
- Some vendors still ship computers with one sys admin account installed with a default password

6) Impersonation attacks by exploiting trusted authentication

- Identification *delegated* to trusted source
- E.g., on Unix with .rhosts/.rlogin (see 4a above)
- Each delegation is a potential security hole!
  - Can you really trust the „trusted" source?

E.g., Host A trusts Host B.

User X on Host B can impersonate User Y from Host B.

g-2. Spoofing (1)

- Spoofing — attacker (or attacker's agent) pretends to be a valid entity *without foiling authentication*
  - **Spoof - 1.** To deceive. [...]

    The American Heritage® Dictionary of the English Language: Fourth Edition. 2000
- Don't confuse spoofing with impersonation
  - Impersonation — attacker *foils authentication* and assumes identity of a valid entity
- Three types of spoofing:

1) Masquerading
2) Session hijacking
3) Man-in-the middle (MITM)

Spoofing (2)

1) Masquerading = a host pretends to be another

- Really: attacker sets up the host (host is attacker’s agent)
- Masquerading - Example 1:
  - Real web site: Blue-Bank.com for Blue Bank Corp.
  - Attacker puts a masquerading host at: BlueBank.com
    - It mimics the look of original site as closely as possible
  - A mistyping user (who just missed „-”) is asked to login, to give password => sensitive info disclosure
  - Can get users to masquerading site by other means
    - E.g., advertise masquerading host with banners on other web sites (banners would just say „Blue Bank”-no „-” there)
- Similar typical masquerades:
  - xyz.org *and* xyz.net masquerade as xyz.com
  - 10pht.com masquerades as lOpht.com (1-I, 0-O)
  - citicar.com masquerades as citycar.com

Spoofing (3)

- Masquerading - Example 2:
  - Attacker exploits web server flaw – modifies web pages
  - Makes no visible changes but „steals" customers
  - E.g., Books-R-Us web site could be changed in a sneaky way:
    - Processing of browsing customers remains unchanged

    BUT
    - Processing of ordering customers modified: (some) orders sent to competing Books Depot
      - Only „some" to mask the masquerade

Spoofing (4)

2) Session hijacking = attacker intercepting and carrying on a session begun by a legitimate entity

- Session hijacking - Example 1
  - Books Depot wiretaps network and intercepts packets
  - After buyer finds a book she wants at Books-R-Us and starts ordering it,

    the order is taken over by Books Depot
- Session hijacking - Example 2
  - Sysadmin starts Telnet session by remotely logging in to his privileged acct
  - Attacker uses hijacking utility to intrude in the session
    - Can send his own commands between admin's commands
    - System treats commands as coming from sysadmin

Spoofing (5)

3) Man-in-the middle (MITM)

*** SKIP "3) Man-in-the middle (MITM)" (this & next slide) – will cover after encryption explained ***

- Similar to hijacking
- Difference: MITM participates in a session from its start

  (session hijacking occurs *after* session established)

...continued....

Spoofing (6)

***** SKIP *****

- MITM – Example: Alice sends encrypted msg to Bob

(a) Correct communication

- Alice requests key distributor for $K_{PUB\text{-}Bob}$
- Key distributor sends $K_{PUB\text{-}Bob}$ to Alice
- Alice encrypts P: $C = E(P, K_{PUB\text{-}Bob})$ & sends C to Bob
- Bob receives C and decrypts it: $P = D(C, K_{PRIV\text{-}Bob})$

(b) MITM attack

- Alice requests key distributor for $K_{PUB\text{-}Bob}$
- MITM intercepts request & sends $K_{PUB\text{-}MITM}$ to Alice
- Alice encr. P: $C = E(P, K_{PUB\text{-}MITM})$ & sends C to Bob
- MITM intercepts C & decrypts it: $P = D(C, K_{PRIV\text{-}MITM})$
- MITM requests key distributor for $K_{PUB\text{-}Bob}$
- Key distributor sends $K_{PUB\text{-}Bob}$ to MITM
- MITM encr. P: $C = E(P, K_{PUB\text{-}Bob})$ & sends C to Bob
- Bob receives C and decrypts it: $P = D(C, K_{PRIV\text{-}Bob})$

Note: Neither Alice not Bob know about MITM attack

g-3. Message confidentiality threats (1)

- Message confidentiality threats include:

1) Eavesdropping – above

2) Impersonation – above

3) Misdelivery
    - Msg delivered to a wrong person due to:
        - Network flaw
        - Human error
            - Email addresses should not be cryptic

              iwalkey@org.com better than iw@org.com

              iwalker@org.com better than 10064,30652@org.com

Message confidentiality threats (2)

4) Exposure

- Msg can be exposed at any moment between its creation and disposal
- Some points of msg exposure:
  - Temporary buffers
  - Switches / routers / gateways / intermediate hosts
  - Workspaces of processes that build / format / present msg (including OS and app pgms)
- Many ways of msg exposure:
  - Passive wiretapping
  - Interception by impersonator at source / in transit / at destination

5) Traffic flow analysis

- Mere existence of msg (even if content unknown) can reveal sth important
  - E.g., heavy msg traffic form one node in a military network might indicate it's headquarters

g-4. Message integrity threats (1)

- Message integrity threats include:
  1) Msg fabrication
  2) Noise

1) Msg fabrication

- Receiver of fabricated msg may be misled to do what fabricated msg requests or demands
- Some types of msg fabrication:
  - Changing part of/entire msg body
  - Completely replacing whole msg (body & header)
  - Replay old msg
  - Combine pieces of old msgs
  - Change apparent msg source
  - Destroy/delete msg

Message integrity threats (2)

- Means of msg fabrication:
  - Active wiretap
  - Trojan horse
  - Impersonation
  - Taking over host/workstation

2) Noise = unintentional interference

- Noise can distort msg
- Communication protocols designed to detect/correct transmission errors
  - Corrected by:
    - error correcting codes
    - retransmission

g-5. Web site attacks (1)

- Web site attacks – quite common due to:
  - Visibility
    - E.g., web site defacement – changing web site appearance
  - Ease of attack
    - Web site code available to attacker (Menu: View>>Source)
    - A lot of vulnerabilities in web server s/w
      - E.g., 17 security patches for MS web server s/w, IIS v. 4.0 in 18 months

- Common Web site attacks (discussed next):

1) Buffer overflows
2) Dot-dot attacks
3) Exploiting application code errors
4) Server-side include

Web site attacks (2)

1) Buffer overflows

- Attacker feeds pgm much more data than it expects
  - WILL BE DISCUSSED in the "Program Security" Chapter
- iishack - best known web server buffer overflow problem
  - Procedure executing this attack is available

Web site attacks (3)

2) Dot-dot attacks

- In Unix & Windows: '..' points to parent directory
- Example attack: on webhits.dll for MS Index Server
  - Pass the following URL to the server

http://URL/null.htw?CiWebHitsFile=/../../../../../winnt/system32/autoexec.nt

  - Returns *autoexec.nt* file – attacker can modify it
- Other example attacks: Lab Manual – p. 257
  - Using ..%255c.. in URL allows executing arbitrary commands
- Solution to (some) dot-dot attacks:

  1) Have no editors, xterm, telnet, utilities on web server

     => no s/w to be executed by an attacker on web server to help him

  2) Create a fence confining web server

3) Exploiting application code errors

- Source of problem:
  - Web server may have k*1,000 transactions at a time
  - Might use *parameter fields* (appended to URL) to keep track of transaction status
- Example: exploiting *incomplete mediation* in app (cf. earlier)
  - URL generated by *client's browser* to access web server, e.g.:

http://www.things.com/order/final&custID=101&part=555A&qy=20&price=10&ship=boat&shipcost=5&total=205

  - Instead, *user* edits URL directly, changing price and total cost as follows:

http://www.things.com/order/final&custID=101&part=555A&qy=20&price=1&ship=boat&shipcost=5&total=25

  - User sends forged URL to web server
    - The server takes 25 as the total cost

Web site attacks (5)

4) Server-side include

- HTML code for web page can contain *include* commands
- Example
  - Attacker can open telnet session on server (with server's privileges)
  - <!-#exec cmd=/"usr/bin/telnet &"->
- *include exex* (# exec) commands can be used to execute an arbitrary file on the server
- Attacker can execute, e.g., commands such as:
  - chmod – changes access rights
  - sh – establish command shell
  - cat – copy to a file

g-6. Denial of service (attack on avail.) (1)

- Service can be denied:

A) due to (nonmalicious) failures

- Examples:
  - Line cut accidentally (e.g., by a construction crew)
  - Noise on a line
  - Node/device failure (s/w or h/w failure)
  - Device saturation (due to nonmalicious excessive workload/ or traffic)
- Some of the above service denials are short-lived and/or go away automatically (e.g., noise, some device saturations)

B) due to denial-of-service (DoS) attacks = attacks on *availab.*

- DoS attacks include:
  1) Physical DoS attacks
  2) Electronic DoS attacks

Denial of service (2)

1) Physical DoS attacks – examples:

- Line cut deliberately
- Noise injected on a line
- Bringing down a node/device via h/w manipulation

2) Electronic DoS attacks – examples:

(2a) Crashing nodes/devices via s/w manipulation

- Many examples discussed earlier

(2b) Saturating devices (due to malicious injection of excessive workload/ or traffic)

Includes:

(i) Connection flooding

(ii) SYN flood

(2c) Redirecting traffic

Includes:

(i) Packet-dropping attacks (incl. black hole attacks)

(ii) DNS attacks

Denial of service (3) – 2b: Saturating devices – i: Connection flooding

(i) Connection flooding

= flooding a connection with useless packets so it has no capacity to handle (more) useful packets

- ICMP (Internet Control Msg Protocol) - designed for Internet system diagnostic (3rd class of Internet protocols next to TCP/IP & UDP)

  ICMP msgs can be used for attacks

  - Some ICMP msgs:

    - *echo request* – source S requests destination D to return data sent to it (shows that link from S to D is good)
    - *echo reply* – response to echo request sent from D to S
    - *destination unreachable* – msg to S indicating that packet can't be delivered to D
    - *source quench* – S told to slow down sending msgs to D (indicates that D is becoming saturated)

Note: *ping* sends ICMP „echo request" msg to destination D.

If D replies with „echo reply" msg, it indicates that D is reachable/functioning (also shows msg round-trip time).

Denial of service (4) – 2b: Saturating devices – i: Connection flooding

Note: Try ping/echo on MS Windows:
(1) Start>>All Programs>>Accessories>>Command Prompt
(2) ping www.wmich.edu (try: www.cs.wmich.edu, cs.wmich.edu)

- Example attacks using ICMP msgs

  (i1) Echo-chargen attack

  - *chargen* protocol – generates stream of packets; used for testing network

  - Echo-chargen attack example 1:

    (1) attacker uses chargen on server X to send stream of *echo request* packets to Y

    (2) Y sends *echo reply* packets back to X

    This creates endless „busy loop" beetw. X & Y

  - Echo-chargen attack example 2:

    (1) attacker uses chargen on X to send stream of *echo request* packets *to X*

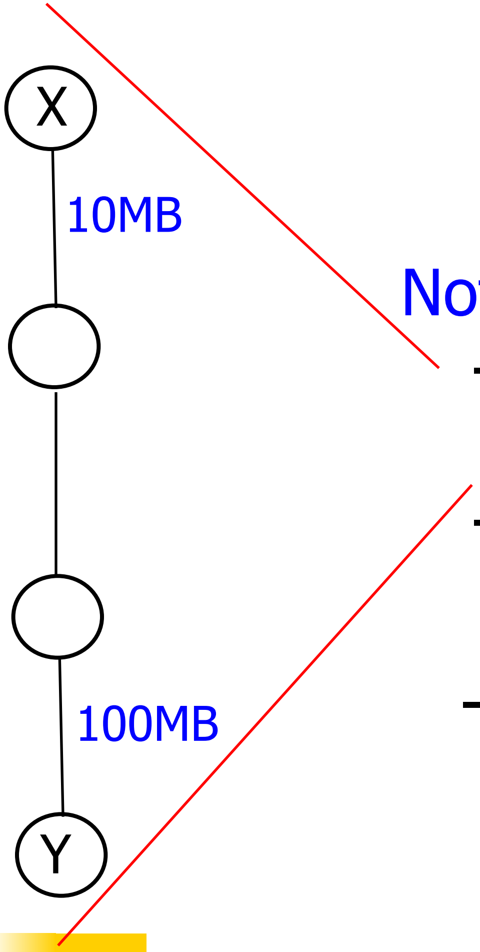
    (2) X sends *echo reply* packets back to itself

Denial of service (5) – 2b: Saturating devices – i: Connection flooding

(i2) Ping of death attack, incl. smurf attack

- Ping of death example :
  (1) attacker uses ping after ping on X to flood Y with pings (ping uses ICMP echo req./reply)
  (2) X responds to pings (to Y)

This creates endless „busy loop" beetw. X & Y



Note: In cases (i1-ex.1) & (i2):

- if X is on 10 MB connection and path to victim Y is 100 MB, X can't flood Y
- if X is on 100 MB connection and path to victim Y is 10 MB, X can easily flood Y

- Smurf attack example:
  (1) attacker spoofs source address of ping packet sent fr. X – appears to be sent by Z
  (2) att. broadcasts spoofed pkt to N hosts
  (3) all N hosts echo to Z – flood it

Denial of service (6) – 2b: Saturating devices – ii: SYN flooding

(ii) SYN flood DoS attack

- Attack is based on properties/implementation of a *session* in TCP protocol suite
- *Session* = virtual connection between protocol peers
  - Session established with *three-way handshake* (S = source, D = destination) as follows:
    - S to D: SYN
    - D to S: SYN+ACK
    - S to D: ACK
    - Now session between S and D is established
  - D keeps *SYN_RECV queue* which tracks connections being established for which it has received no ACK
    - Normally, entry is in SYN_RECV for a short time
    - If no ACK received within time T (usu. a few minutes), entry discarded (connection establ. times out)

Denial of service (7) – 2b: Saturating devices – ii: SYN flooding

    - Normally, size of SYN_RECV (10-20) is sufficient to accommodate all connections under establishment

- SYN flood attack scenario
  - Attacker sends many SYN requests to D (as if starting 3-way handshake)
  - Attacker never replies to D's SYN+ACK packets
  - D puts entry for each unanswered SYN+ACK packet into SYN_RECV queue
  - With many unanswered SYN+ACK packets, SYN_RECV queue fills up
  - When SYN_RECV is full, no entries for legitimate unanswered SYN+ACK packets can be put into SYN_RECV queue on D

    => nobody can establish legitim. connection with D

Denial of service (8) – 2b: Saturating devices – ii: SYN flooding

- Modification 1 of SYN flood attack scenario: attacker spoofs sender's address in SYN packets sent to D
  - Question: Why?

Denial of service (9) – 2b: Saturating devices – ii: SYN flooding

- Modification 1 of syn flood attack scenario:
  attacker spoofs sender's address in SYN packets sent to D
  - Question: Why?
  - Answer:
    To mask packet's real source, to cover his tracks

- Modification 2 of SYN flood attack scenario:
  attacker makes each spoofed sender's address in SYN packets different
  - Question: Why?

Denial of service (10) – 2b: Saturating devices – ii: SYN flooding

- …

- Modification 2 of SYN flood attack scenario: attacker makes each spoofed sender's address in SYN packets different
  - Question: Why?
  - Answer:
    If all had the same source, detection of attack would be simpler (too many incomplete connection requests coming from the same source look suspicious)

Denial of service (11) – 2c: Redirecting traffic - i: Advertising false best path

(2c) Redirecting traffic (incl. dropping redirected packets)

(i) Redirecting traffic by advertising a false best path

- Routers find best path for passing packets from S to D
  - Routers advertise their conections to their neighbors (cf. Disemination of routing info - Slide 28; ALSO: P&P, p.380—Routing Concepts + Fig. 7-2)
- Example of traffic redirection attack:
  - Router R taken over by attacker
  - R advertises (falsely) to all neighbors that it has the best (e.g., shortest) path to hosts H1, H2, ..., Hn
  - Hosts around R forward to R all packets addressed to H1, H2, ..., Hn
  - R drops *some* or *all* these packets

    drops *some* => packet-dropping attack

    drops *all* => black hole attack

    (black hole attack is spec. case of pkt-drop. attack)

Denial of service (12) – 2c: Redirecting traffic – ii: DNS attacks

(ii) Redirecting traffic by DNS attacks

- Domain name server (DNS)
  - Function: resolving domain name

    = converting domain names into IP addresses
    - E.g., aol.com □ 205.188.142.182
  - DNS queries other DNSs (on other hosts) for info on unknown IP addresses
  - DNS caches query replies (addresses) for efficiency

- Most common DNS implementation:

  BIND s/w (BIND = Berkeley Internet Name Domain)

  a.k.a. *named* (named = name daemon)
  - Numerous flaws in BIND
    - Including buffer overflow

- Attacks on DNS (e.g., on BIND)
  - Overtaking DNS / fabricating cached DNS entries
    - Using fabricated entry to redirect traffic

g-7. Distributed denial of service (attack on availability)

[Fig. courtesy of B. Endicott-Popovsky]

- DDoS = distributed denial of service
- Attack scenario:

1) Stage 1:
    - Attacker plants Trojans on many target machines
        - Target machines controlled by Trojans become *zombies*

2) Stage 2:
    - Attacker chooses victim V, orders zombies to attack V
    - Each zombie launches a separate DoS attack
        - Different zombies can use different DoS attacks
            - E.g., some use syn floods, other smurf attacks
            - This probes different weak points
        - All attacks together constitute a DDoS
    - V becomes overwhelmed and unavailable

=> DDoS succeeds