

# CSCE 727

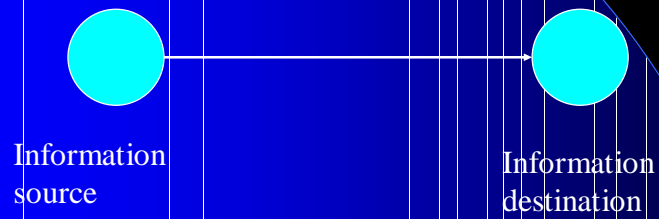
## Cyber Attacks

# Attack

RFC 2828:

“ An assault on system security that derives from an intelligent threat, i.e., an intelligent act that is a deliberate attempt (especially in the sense of a method or technique) to evade security services and violate the security policy of the system.”

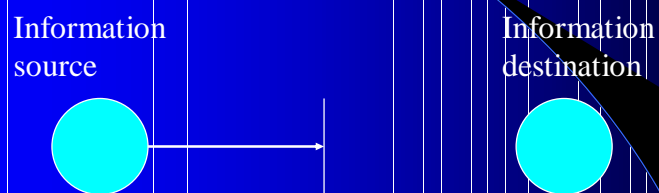
## Normal Flow



CSCE 727 - Farkas

3

## Interruption



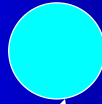
Asset is **destroyed** or becomes **unavailable** - **Availability**  
**Example:** destruction of hardware, cutting communication line, disabling file management system, etc.

CSCE 727 - Farkas

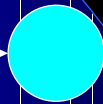
4

## Interception

Information  
source



Information  
destination



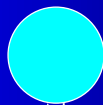
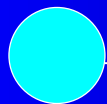
**Unauthorized** party gains **access** to the asset – **Confidentiality**  
Example: wiretapping, unauthorized copying of files

CSCE 727 - Farkas

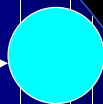
5

## Modification

Information  
source



Information  
destination



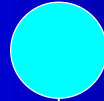
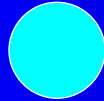
**Unauthorized** party **tampers** with the asset – **Integrity**  
Example: changing values of data, altering programs, modify  
content of a message, etc.

CSCE 727 - Farkas

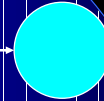
6

## Fabrication

Information source



Information destination



Unauthorized party **insets counterfeit object** into the system –

**Authenticity**

**Example:** insertion of offending messages, addition of records to a file, etc.

CSCE 727 - Farkas

7

## Phases of Attack

Improve detection by examining which “phase” an intruder’s behavior is identified

Attack phases:

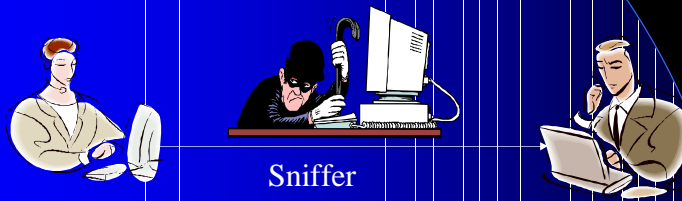
- Intelligence gathering: attacker observes the system to determine vulnerabilities
- Planning: attacker decide what resource to attack (usually least defended component)
- Attack: attacker carries out the plan
- Inside the system:
  - | Hiding: attacker covers tracks of attack
  - | Future attacks: attacker installs backdoors for future entry points

CSCE 727 - Farkas

8

## Passive Attack

“Attempts to learn or make use of information from the system but does not affect system resources” (RFC 2828)



## Sniffers

- | All machines on a network can “hear” ongoing traffic
- | A machine will respond only to data addressed specifically to it
- | Network interface: “promiscuous mode” – able to **capture all frames** transmitted on the local area network segment

# Risks of Sniffers

- | Serious security threat
- | Capture confidential information
  - Authentication information
  - Private data
- | Capture network traffic information

Passive attacks

Interception (confidentiality)

Release of message contents

Traffic analysis

## Release of message content

- | Intruder is able to **interpret** and **extract information** being transmitted
- | Highest risk: **authentication information**
  - Can be used to compromise additional system resources

## Traffic Analysis

- | Intruder is **not able** to **interpret** and **extract** the transmitted information
- | Intruder is able to **derive** (infer) **information** from the traffic characteristics

## Protection against passive attacks

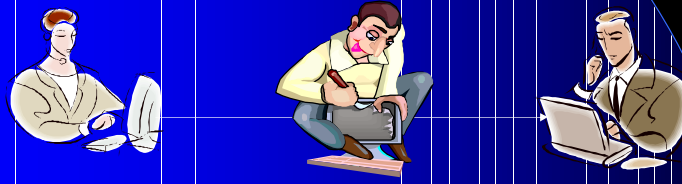
- | Shield confidential data from sniffers: cryptography
- | Disturb traffic pattern: NRL
  - Traffic padding
  - Onion routing
- | Modern switch technology: network traffic is directed to the destination interfaces
- | **Detect and eliminate sniffers**

## Detection of Sniffer Tools

- | **Difficult to detect:** passive programs
- | Tools:
  - **Sniffest** – SunOS and Solaris: can detect sniffers even if the network interface is not in promiscuous mode
  - **Nitwitt** – Network Interface Tap: can detect sniffers even if the network interface is not in promiscuous mode
  - **Promisc** – Linux
  - **cmp** – SunOS 4.x: detects promiscuous mode
- | **AntiSniff** (L0pht Heavy Industries, Inc. ): remotely detects computers that are packet sniffing, regardless of the OS

# Active attacks

“Attempts to alter system resources or affect their operation” (RFC 2828)



## Active attacks

Interruption  
(availability)

Modification  
(integrity)

Fabrication  
(integrity)

## Active Attacks

- | Masquerade
- | Replay
- | Modification of messages
- | Denial of service
- | Degradation of service
- | Spoofing attacks
- | Session hijacking

## Masquerade

- | One entity pretends to be a different entity
- | Usually involves additional attacks, e.g.,
  - Password Crack
  - Authentication sequences captured and replay

## User Identification Techniques

- | What the claimant knows
  - Password, personal information
- | What the claimant possesses
  - Physical key, ticket, passport, token, smart card
- | What the claimant is (biometrics)
  - Fingerprints, voiceprint, signature dynamics
- | Where the claimant is
  - Network address, physical location

## Passwords

- | Commonly used method
- | For each user, system stores (user name,  $F(\text{password})$ ), where  $F$  is some transformation (e.g., one-way cipher) in a password file
  - $F(\text{password})$  is easy to compute
  - From  $F(\text{password})$ , password is difficult to compute
  - Password is not stored in the system
- | When user enters the password, system computes  $F(\text{password})$ ; match provides proof of identity

## Vulnerabilities of Passwords

- | Inherent vulnerabilities
  - Easy to guess or snoop
  - No control on sharing
- | Practical vulnerabilities
  - Visible in the clear in distributed and network environment
  - Susceptible for replay attacks if encrypted naively

## Weak Passwords

- | Bell Labs study (Morris and Thompson, 1979), 3289 passwords were examined
  - 15 single ASCII characters, 72 two ASCII characters, 464 three ASCII characters, 477 four ASCII characters, 706 five letters (all lower case or all upper case), 605 six letters, all lower case, 492 weak passwords (name, dictionary words, etc.)
  - Summary: 2831 passwords (86% of the sample) were weak, i.e., either too easy to predict or too short

## Dictionary Attacks on Passwords

### Attack 1:

- Create dictionary of common words and names and their simple transformations
- Use these to guess password

### Attack 2:

- Usually  $F$  is public and so is the password file (encrypted)
- Compute  $F(\text{word})$  for each word in dictionary
- Find match

### Attack 3:

- Pre-compute dictionary
- Look up matches

## Password Salt

Used to make dictionary attack more difficult

Salt is a 12 bit number between 0 and 4095

It is derived from the system clock and the process identifier

Compute  $F(\text{password}+\text{salt})$ ; both salt and  $F(\text{password}+\text{salt})$  are stored in the password table

User: gives password, system finds salt and computes  $F(\text{password}+\text{salt})$  and check for match

Note: with salt, the same password is computed in 4096 ways

## Password Management Policy

- | Educate users to make better choices
- | Define rules for good password selection and ask users to follow them
- | Ask or force users to change their password periodically
- | Actively attempt to break user's passwords and force users to change broken ones
- | Screen password choices

## Replay

- | Passive capture of data unit and its retransmission

## Modification of messages

- | Some portion of the legitimate message is altered or
- | Message is delayed or reordered

## Denial of service

- | Prevents or inhibits the normal use or management of resources
- | May range from blocking a particular resource or the entire network
- | Past attacks: aim to crash systems of a victim

## DoS attacks

| **E-mail bombing attack:** floods victim's mail with large bogus messages

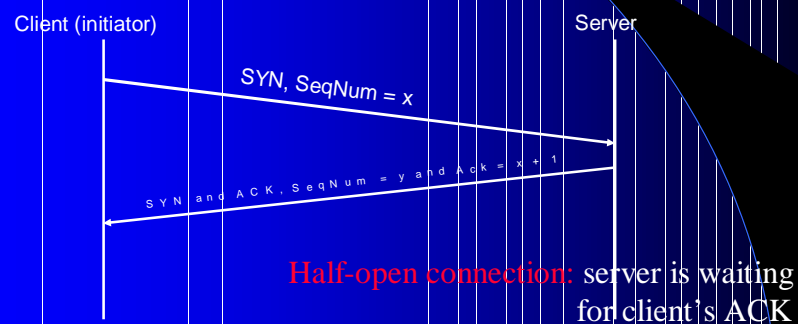
- Popular
- Free tools available

| **Smurf attack:**

- Attacker multicast or broadcast an Internet Control Message Protocol (ICMP) with spoofed IP address of the victim system
- Each receiving system sends a respond to the victim
- Victim's system is flooded

## DoS attacks

| **TCP SYN flooding**



## TCP SYN flooding

- | Server: limited number of allowed half-open connections
- | Backlog queue:
  - Existing half-open connections
  - Full: no new connections can be established
  - Time-out, reset

## TCP SYN flooding

- | Attack:
  - Attacker: send SYN requests to server with IP source that unable to response to SYN-ACK
  - Server's backlog queue filled
  - No new connections can be established
  - Keep sending SYN requests
- | Does not affect
  - Existing or open incoming connections
  - Outgoing connections

## Distributed denial of service (DDoS)

- | Use additional systems (“zombies”) on the Internet to launch a coordinated attack

## Protection against DoS, DDoS

- | Hard to provide full protection
- | Some of the attacks can be prevented
  - Filter out incoming traffic with local IP address as source
  - Avoid established state until confirmation of client’s identity
- | Internet trace back: determine the source of an attack

## Degradation of Service

- | Do not completely block service just reduce the quality of service

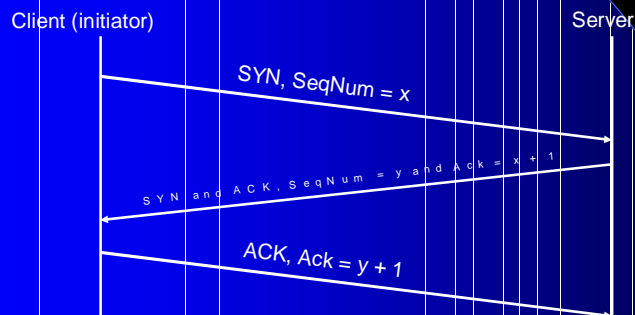
## Spoofing attacks

- | IP spoofing
- | DNS spoofing
- | Sequence number guessing

# Sequence number guessing

## Weaknesses:

- | TCP/IP host does not verify the authenticity of the source IP
- |  $x, y$  are not randomly generated => attacker may guess value of  $y$  with good accuracy



# Intrusion Control

It is better to prevent something than to plan for loss.

**Problem: Misuse happens!**

## Need:

- | Intrusion Prevention: protect system resources
- | Intrusion Detection: (second line of defense) identify misuse
- | Intrusion Recovery: cost effective recovery models

## Intrusion Prevention

- | First line of defense
- | Techniques: cryptography, identification, authentication, authorization, access control, security filters, etc.
- | Not good enough (prevention, reconstructions)

## Intrusion Detection System (IDS)

- | Looks for specific patterns (attack signatures or abnormal usage) that indicate malicious or suspicious intent
- | Second line of defense against both internal and external threats

## Intrusion Detection Systems

- | Deter intruders
- | Catch intruders
- | Prevent threats to fully occur (real-time IDS)
- | Improve prevention techniques
- | IDS deployment, customisation and management is generally not trivial

## Intrusion Detection - Milestones

- | **1980:** Deviation from historical system usage (Anderson)
- | **1987:** framework for general-purpose intrusion detection system (Denning)
- | **1988:** intrusion detection research divided
  - Attack signatures based detection (MIDAS)
  - Anomaly detection based detection (IDES)

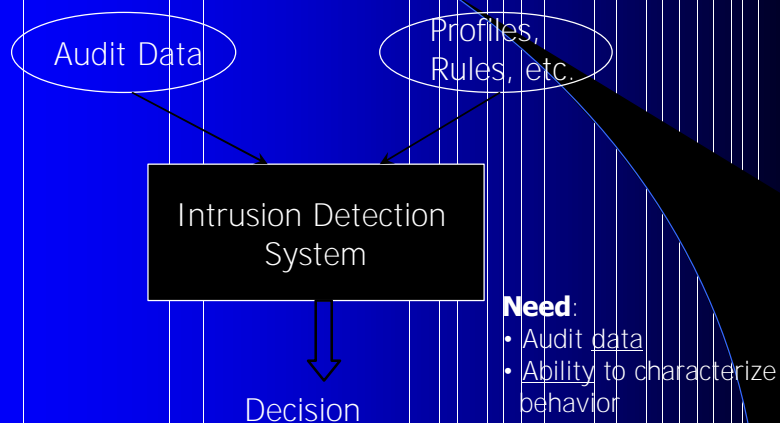
## Intrusion Detection - Milestones

- | **Early 1990s:** Commercial installations
  - IDES, NIDES (SRI)
  - Haystack, Stalker (Haystack Laboratory Inc.)
  - Distributed Intrusion Detection System (Air Force)
- | **Late 1990s - today:**
  - Integration of audit sources
  - Network based intrusion detection
  - Hybrid models
  - Immune system based IDS

# Terminology

- | Audit: activity of looking at user/system behavior, its effects, or the collected data
- | Profiling: looking at users or systems to determine what they usually do
- | Anomaly: abnormal behavior
- | Misuse: activity that violates the security policy
- | Outsider: someone without access right to the system
- | Insider: someone with access right to the system
- | Intrusion: misuse by outsiders and insiders

# Audit-Based Intrusion Detection



## Audit Data

- | Format, granularity and completeness depend on the collecting tool
- | Examples
  - System tools collect data (login, mail)
  - Additional collection of low system level
  - “Sniffers” as network probes
  - Application auditing
- | Needed for
  - Establishing guilt of attackers
  - Detecting suspicious user activities

## Audit Data Accuracy

- | Collection method
  - System architecture and collection point
  - Software and hardware used for collection
- | Storage method
  - Protection of audit data
- | Sharing
  - Transmission protection and correctness
  - Availability

## IDS Categories

1. Time of data analysis
  - | *Real-time v.s. off-the-line IDS*
2. Location where audit data was gathered
  - | *Host-based v.s. network-based v.s. hybrid*
3. Technique used for analysis
  - | *Rule-based v.s. statistic-based*
4. Location of analysis
  - | *Centralized, distributed, network-based*
5. Pattern IDS looking for
  - | *Misuse v.s. anomaly-based v.s. hybrid*

## Commercial IDS

- *Bro* and *Snort* – open source public-domain system
- ISS – *Real Secure* from Internet Security Systems:
  - Real time IDS.
  - Contains both host and network based IDS.
- Tripwire – File integrity assessment tool

## Intrusion Recovery

- Actions to avoid further loss from intrusion
- Terminate intrusion and protect against reoccurrence
- Law enforcement
- Enhance defensive security
- Reconstructive methods based on:
  - Time period of intrusion
  - Changes made by legitimate users during the effected period
  - Regular backups, audit trail based detection of effected components, semantic based recovery, minimal roll-back for recovery.

## What is “Survivability”?

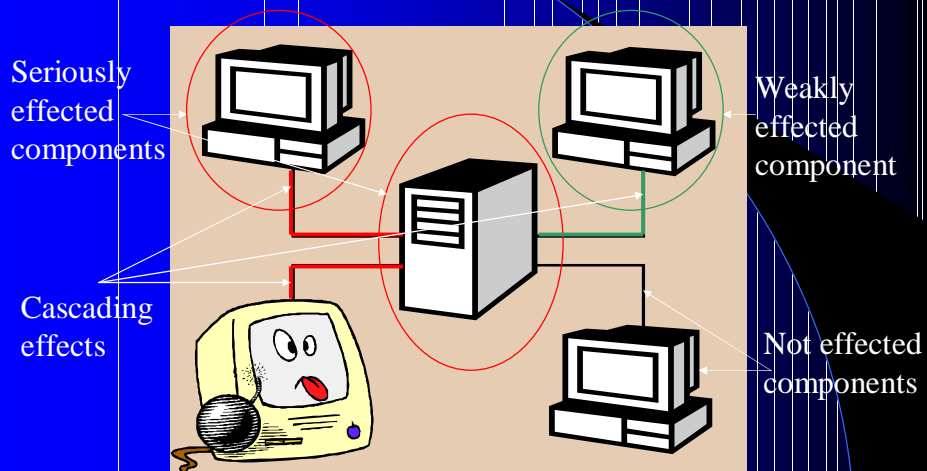
To decide whether a computer system is “survivable”, you must first decide what “survivable” *means*.



# Vulnerable Components

1. Hardware
2. Software
3. Data
4. Communications
5. People

# Effect Modeling and Vulnerability Detection



# Robust System Development

Effects and system dependencies à cascading effects

Cascading and escalating effect modeling à vulnerabilities

Vulnerabilities and their priorities à reduce vulnerabilities: installing safeguards, reconstruct network, redundancy, etc.

Reduced vulnerabilities à estimation of components' security (reliability, correctness, trustworthiness)

Estimation of components' security: cost effective dynamic network resource allocations