WILLIAM & MARY

CSCI 454/554 Computer and Network Security

Topic 8.4 Firewalls and Intrusion Detection Systems (IDS)

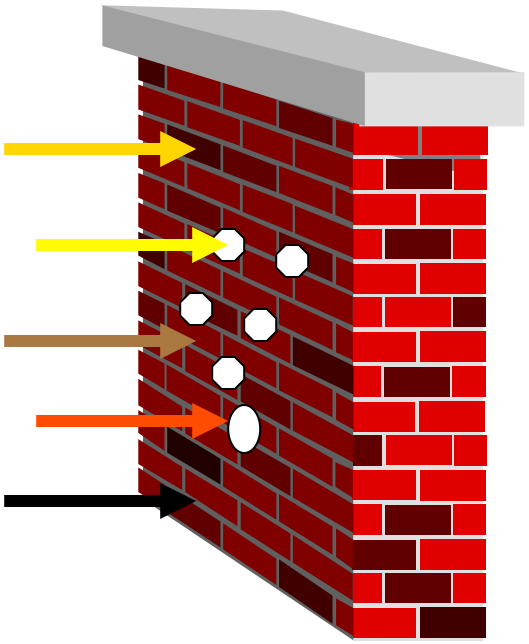
Outline

WILLIAM & MARY

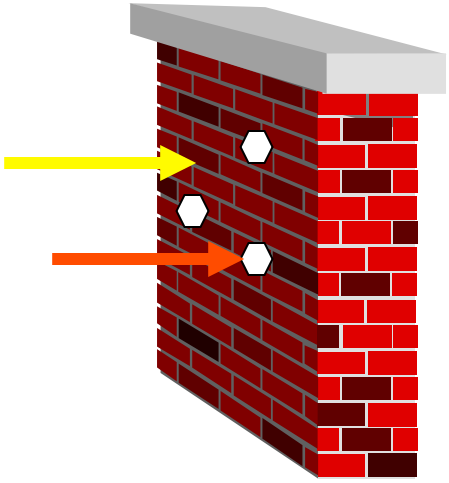
- Firewalls
  - Filtering firewalls
  - Proxy firewalls
- Intrusion Detection System (IDS)
  - Rule-based IDS
  - Anomaly detection IDS
  - Host-based vs. network-based IDS

Overview of Firewalls

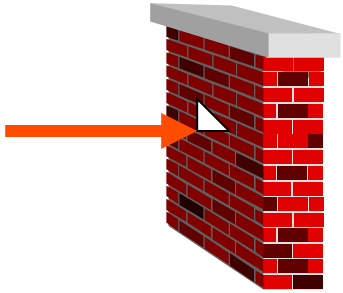
Internet Security Mechanisms



Prevent:
Firewall, IPsec, SSL



Detect:
Intrusion Detection



Survive/
Response:
Recovery, Forensics

- Goal: prevent if possible; detect quickly otherwise; and confine the damage

Basic Terms

WILLIAM & MARY

- *Vulnerabilities*
- *Intrusion*s (attacks) and Intrusion Detection Systems (IDS)
- *Alert or alarm*: message generated by IDS

Example Attacks

WILLIAM & MARY

- Disclosure, modification, and destruction of data
- Compromise host and then attack other systems
- Monitoring and capture of user passwords, then masquerade as authorized user
- Phishing attacks

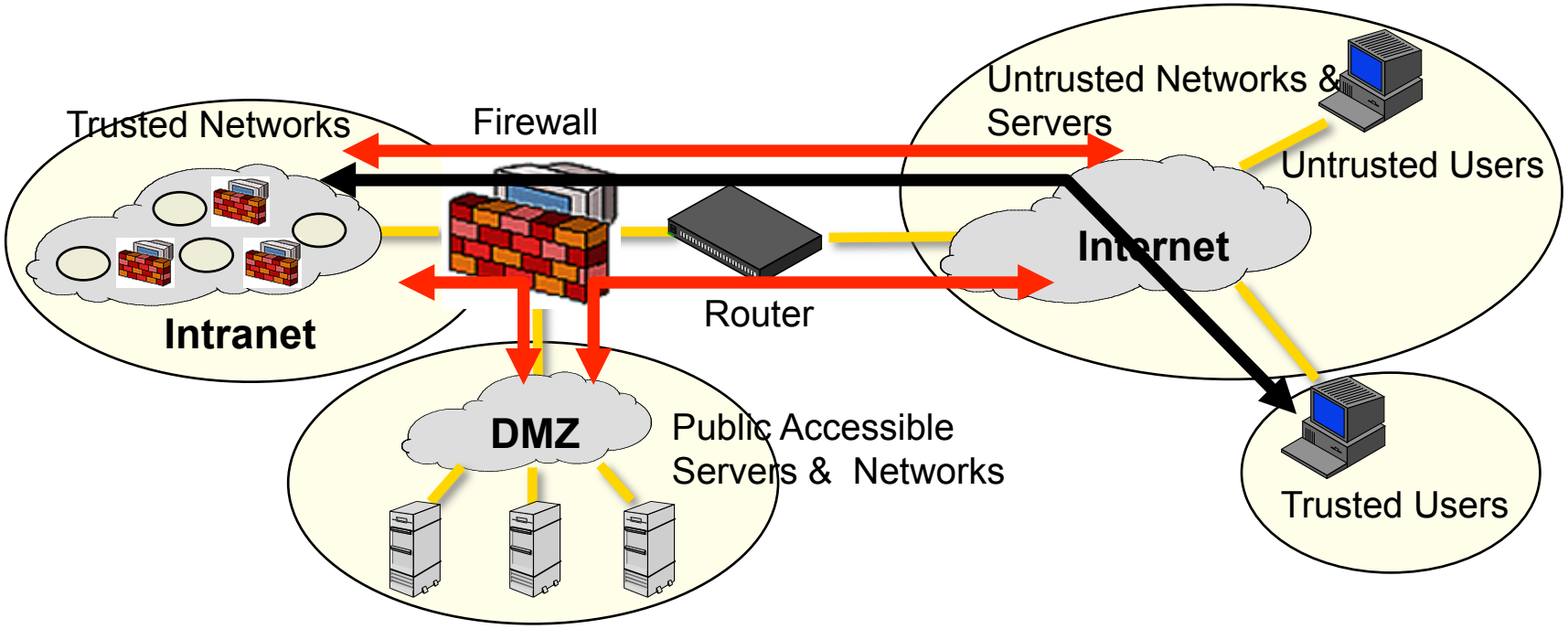
Firewalls

WILLIAM & MARY

- Provides secure connectivity between networks
- Implements and enforces a security policy for communication between networks

Firewalls (Cont'd)

- Many organizations have distinct needs
  - access by anyone to public data concerning the company
  - access only by employees to internal data
- Solution: inner and outer (DMZ) networks



Firewall Capabilities

WILLIAM & MARY

- Controlled access
  - restrict incoming and outgoing traffic according to security policy
- Other functions
  - log traffic, for later analysis
  - network address translation
  - encryption / decryption
  - application (payload) transformations

Limitations of Firewalls

WILLIAM & MARY

- Cannot protect against traffic that does not cross it
  - i.e., there may be other ingress points to the network, such as modems or wireless access points, that bypass the firewall
  - doesn't protect against "inside" attacks
- Configuration of firewalls to accomplish a desired high-level security policy is non-trivial

Filtering and Proxy Firewalls

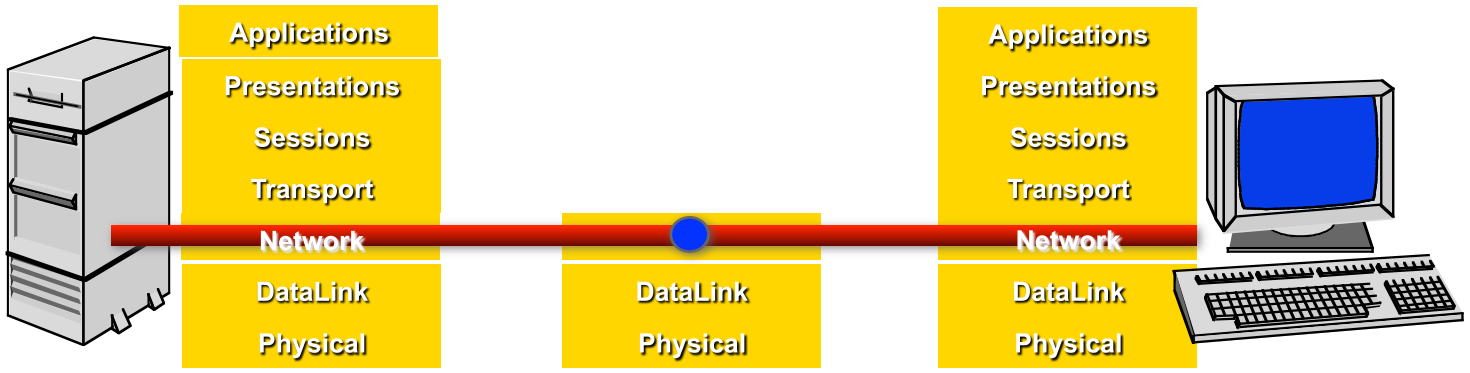
Filtering

WILLIAM & MARY

- Compare traffic to patterns, then process traffic according to rules if matched
- Two styles
  - packet filtering
  - session filtering

Packet Filtering

- Patterns specify values in the header of a single packet, e.g.,
  - source IP address and port number
  - destination IP address and port number
  - transport protocol type



Packet Filtering (cont'd)

WILLIAM & MARY

- Decisions made on a per-packet basis
  - no state information (about previous packets) is maintained or used
- Assessment
  - easy to implement
  - but limited capabilities
- May be subject to tiny-fragment attack
  - first fragment has only a few bytes
  - rest of TCP header in a second fragment, not examined by firewall

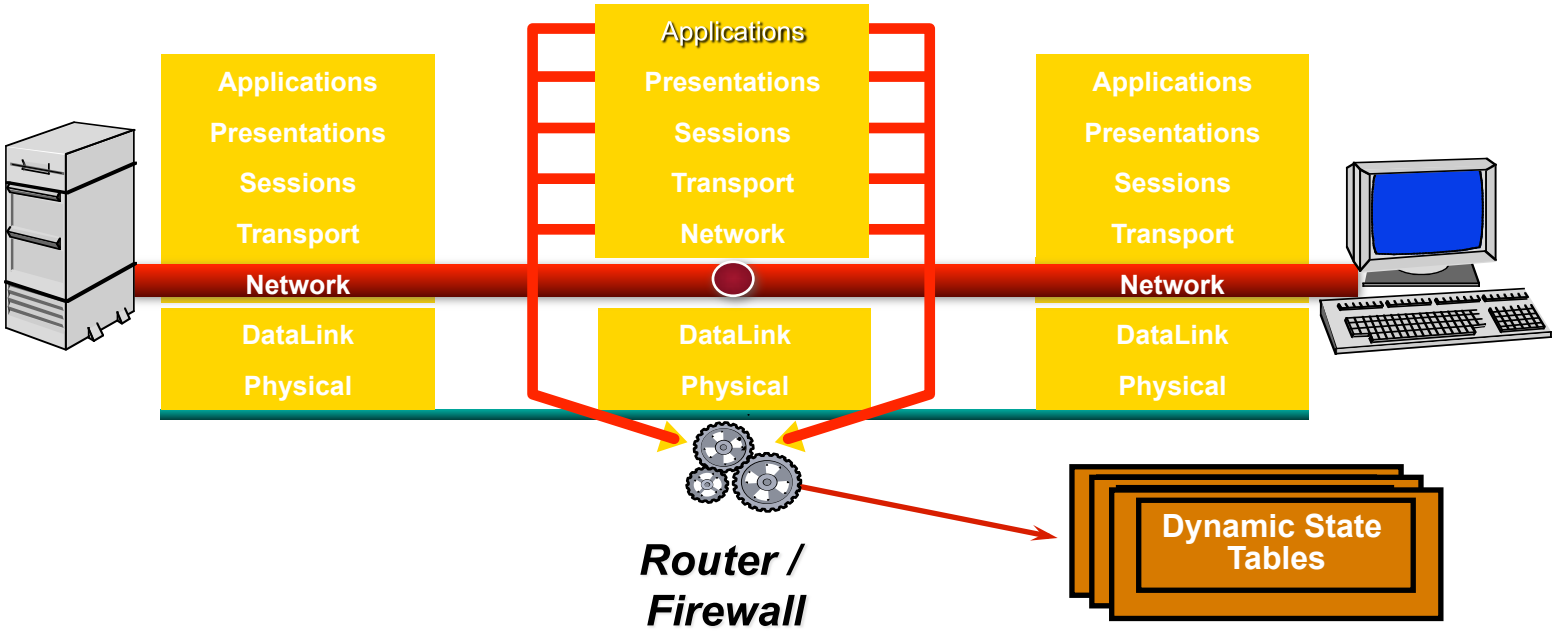
Session Filtering

WILLIAM & MARY

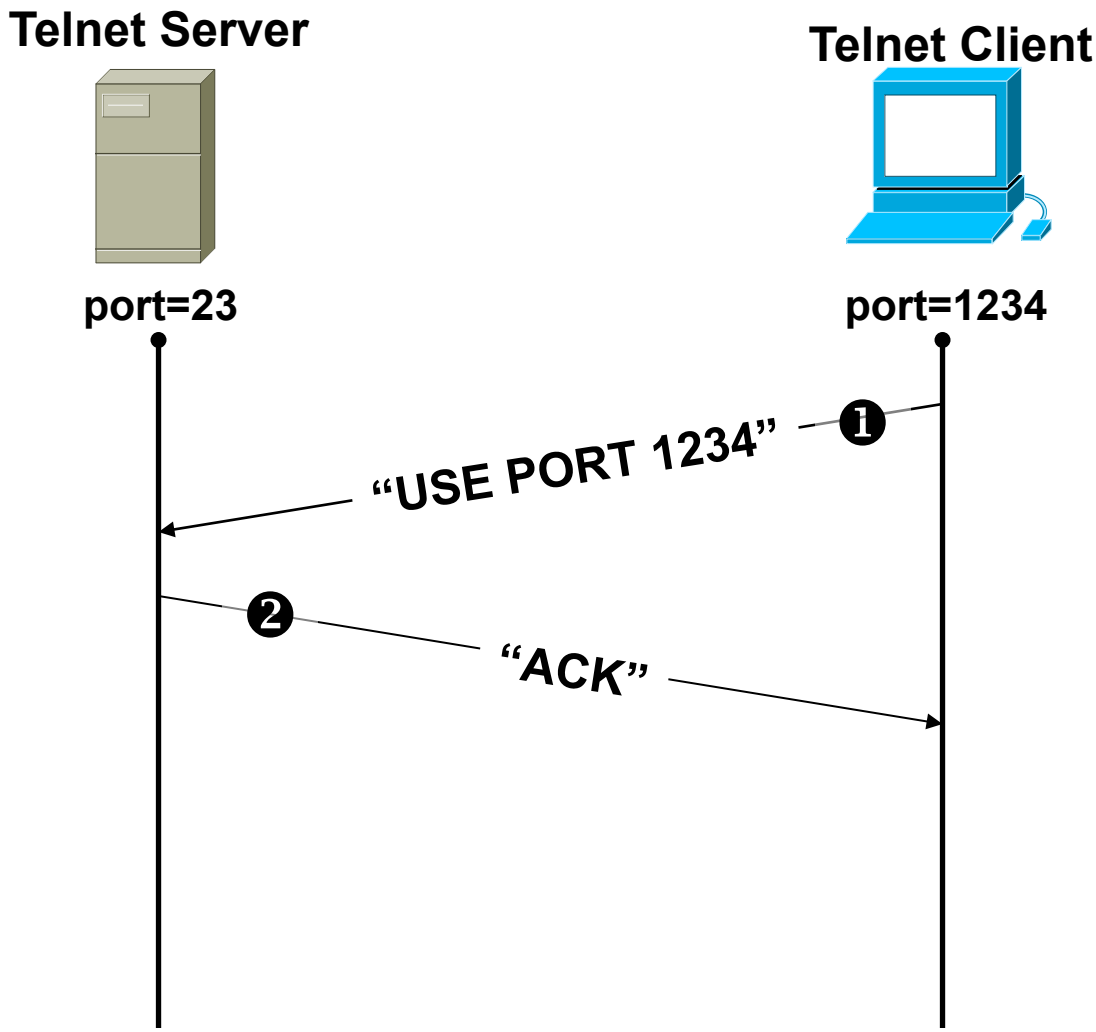
- Packet decisions are made in the context of a *connection* or *flow* of packets
- If packet is the start of a new connection…
  - check against rules for new connections
- If packet is part of an existing connection…
  - check against state-based rules for existing connections
  - update state of this connection

Session Filtering (cont'd)

- Assessment
  - more powerful than packet filtering, can recognize more sophisticated threats or implement more complex policies
  - also more expensive to implement



Application: Telnet



❶ Client opens channel to server; tells server its port number. The ACK bit is not set when initiating the connection but will be set on the remaining packets.

❷ Server acknowleges.

Example: Firewall Access for Telnet

WILLIAM & MARY

Format:

access-list *<rule number>*
<permit|deny>
<protocol>
<SOURCE host with IP address| any|IP address and mask>
[<gt|eq port number>]
<DEST host with IP address| any|IP address and mask>
[<gt|eq port number>]

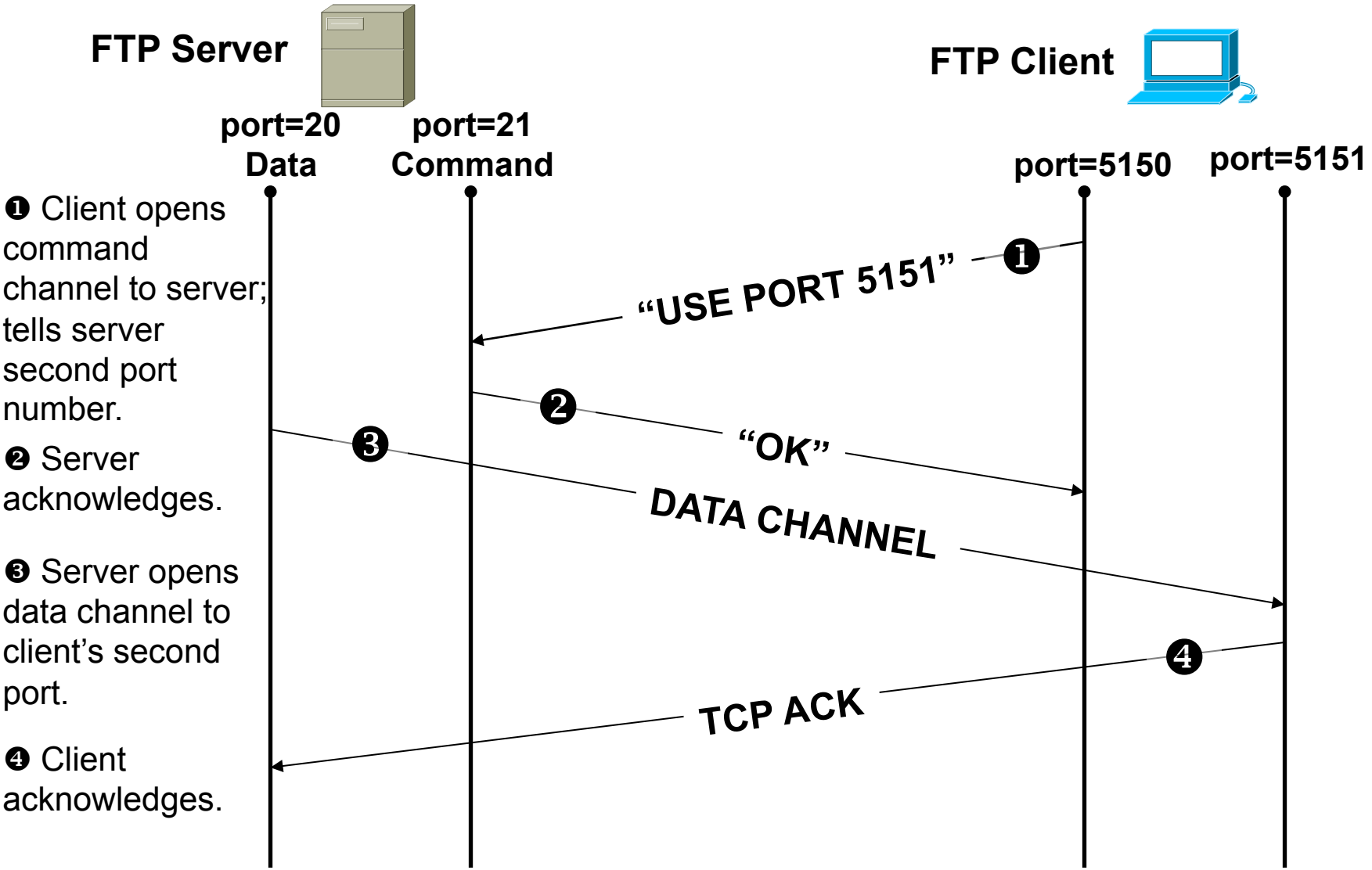
***Note*: any packets not explicitly permitted in an access list assumed to be denied or dropped.**

The following allows user to telnet from an IP address (172.168.10.11) to any destination, but not vice-versa:

```
access-list 100 permit tcp host 172.168.10.11 gt 1023 any eq 23
  ! Allows packets out to remote Telnet servers
access-list 101 permit tcp any eq 23 host 172.168.10.11 established
  ! Allows returning packets to come back in.  It verifies that the ACK bit is set

interface Ethernet 0
 access-list 100 out   ! Apply the first rule to outbound traffic
 access-list 101 in    ! Apply the second rule to inbound traffic
```

Application: FTP



Example: Firewall Access for FTP

WILLIAM & MARY

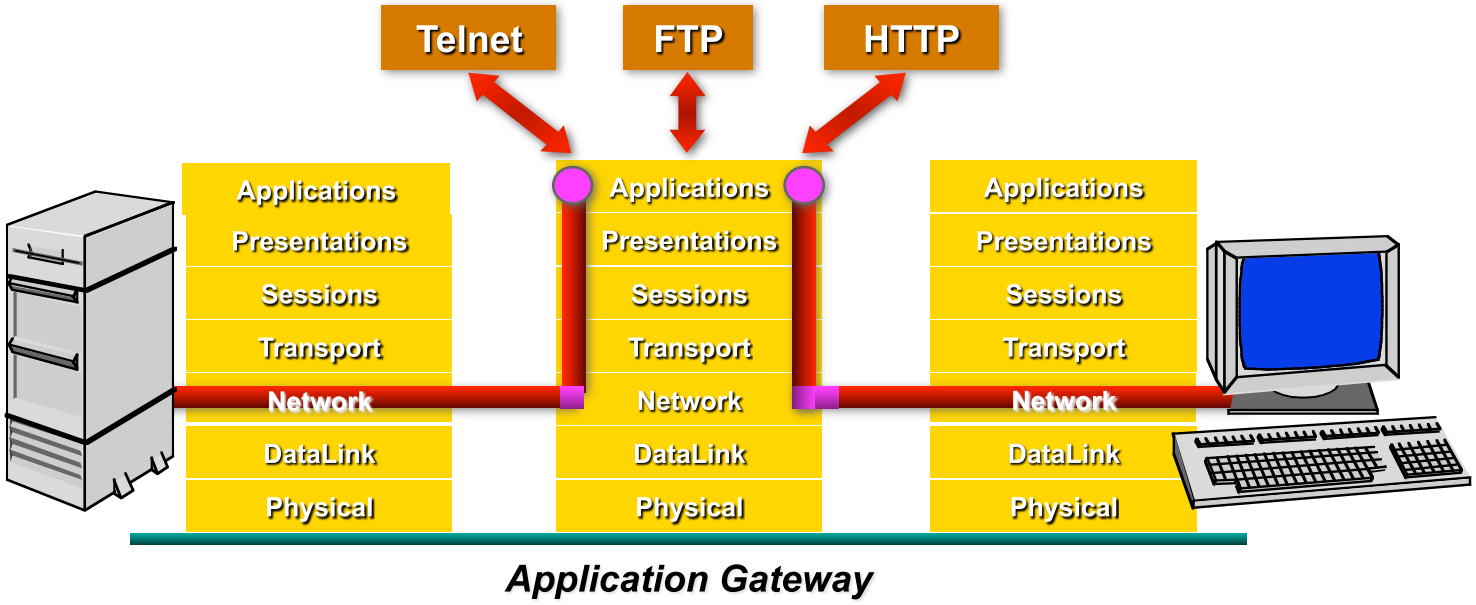
Allow a user to FTP (not passive FTP) from any IP address to the FTP server (172.168.10.12) :

```
access-list 100 permit tcp any gt 1023 host 172.168.10.12 eq 21
access-list 100 permit tcp any gt 1023 host 172.168.10.12 eq 20
  ! Allows packets from any client to the FTP control and data ports
access-list 101 permit tcp host 172.168.10.12 eq 21 any gt 1023
access-list 101 permit tcp host 172.168.10.12 eq 20 any gt 1023
  ! Allows FTP server to send packets back to any IP address with TCP ports > 1023

interface Ethernet 0
 access-list 100 in     ! Apply the first rule to inbound traffic
 access-list 101 out    ! Apply the second rule to outbound traffic
```

Proxy Firewalls

- Serve as *relays* for connections
- Two flavors
  1. application level
  2. circuit level



Application Gateway

Application Proxies

WILLIAM & MARY

- Understand specific application protocols, e.g., HTTP, SMTP, Telnet
  - proxy 'impersonates' both one side of connection to the other
- Can do arbitrary processing / inspection of application payloads
  - ex.: check mail for viruses before forwarding
- Computationally expensive
- Must write a new proxy application to support new protocols

Application Proxies (Cont'd)

- May require hosts inside the organization to be configured to use the proxy

Circuit-Level Proxies

WILLIAM & MARY

- Sets up two connections, one to inside user, one to outside server
  - i.e., proxy at the TCP level, rather than the application level
  - client programs must be aware they are using a circuit-level proxy, by linking to modified libraries
- Users must authenticate to proxy before connection to outside will be established
- Example protocol: SOCKS

Overview of IDS

Attack Stages

WILLIAM & MARY

- Cyber Kill Chain
  - Intelligence gathering: attacker probes the system to determine vulnerabilities
  - Planning: deciding what resource to attack and how
  - Attack execution
  - Hiding: covering traces of the attack
  - Preparation for future attacks: install "back doors" for unhindered access

IDS

WILLIAM & MARY

- Detect if attacks are being attempted, or if system has been compromised
- Desirable features
  - Accuracy
  - Fast
  - Flexible, general
  - Results easy to understand

Measuring Accuracy

WILLIAM & MARY

- *Events* are actions occurring in the system (file accesses, login attempts, etc.)
  - an *intrusion* (I) is an event that is part of an attack
  - an *alarm* (A) is generated if an event is diagnosed as being an intrusion

	Intrusion	**Not an Intrusion**
Alarm Generated	True positive	False positive
Alarm Not Generated	False negative	True negative

Measuring (Cont'd)

WILLIAM & MARY

- True positive rate (TPR): fraction of intrusions correctly diagnosed (detected)
- False negative rate: fraction of intrusions incorrectly diagnosed (not detected)
  - FNR = 1 - TPR
- True negative rate (TNR): fraction of non-intrusions correctly diagnosed
- False positive rate: fraction of non-intrusions incorrectly diagnosed
  - FPR = 1 - TNR

Which Ones Count

WILLIAM & MARY

- It's trivial to have 100% TPR, and trivial to have 0% FPR
  - how?
- Needed: both

Example

WILLIAM & MARY

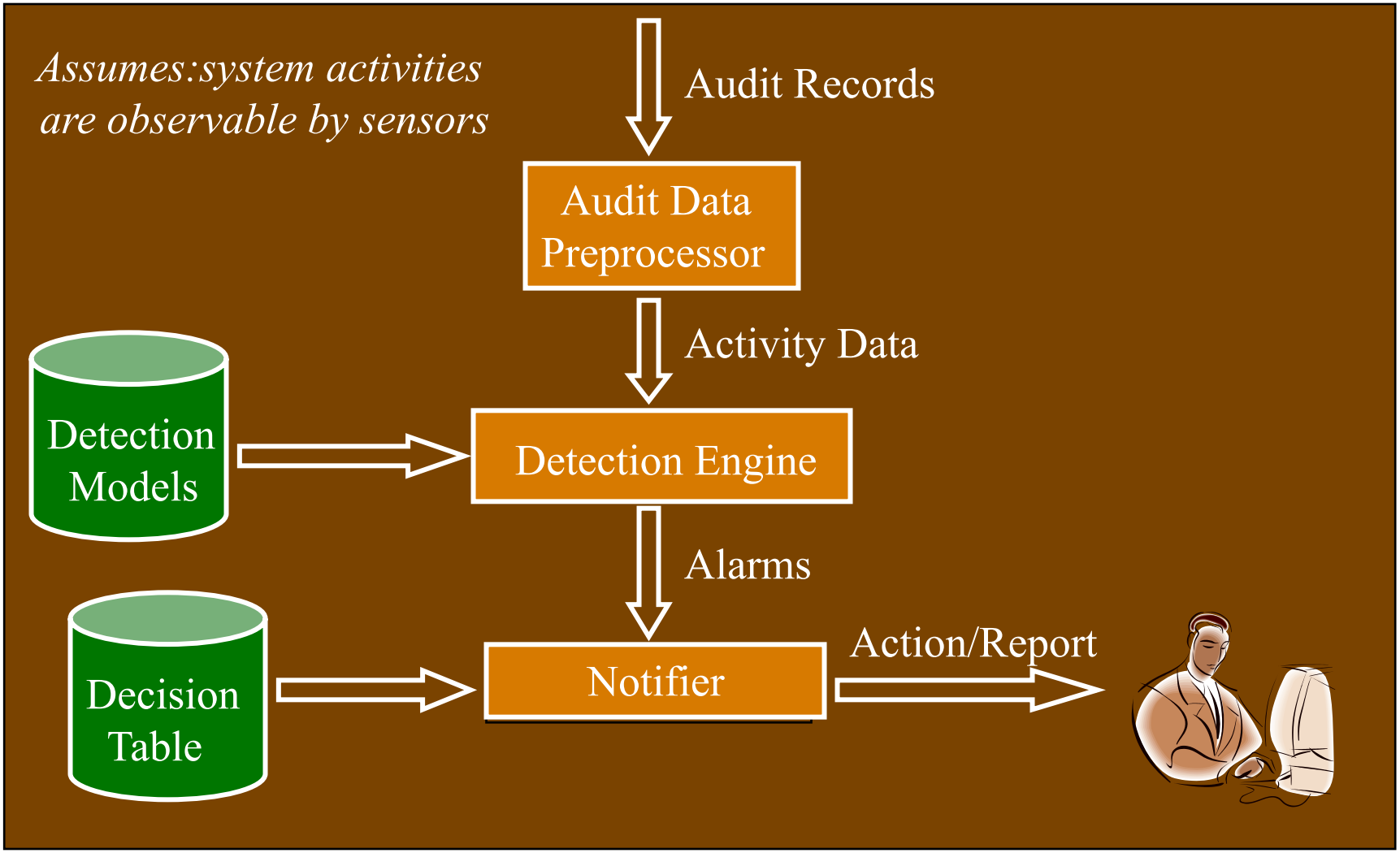
- 70,000 events, 300 intrusions, 2800 alarms (of which 298 are correct diagnoses, 2502 are incorrect)
- TPR: 298 / 300 = 99.3%
- FNR: 0.7%
- TNR: (70000 - 300 - 2502) / (70000 - 300) = 96.4%
- FPR: 3.6%

"Base-Rate Fallacy" Illustrated

WILLIAM & MARY

- IDS often suffers from base-rate fallacy
  - intrusions are rare events
  - non-intrusions are common
  - correctly detected intrusions are swamped by incorrectly detected non-intrusions!
- Previous example: only 298 out of 2800 alarms (10.6%) are correct
  - in reality, often less than 1% of alarms are real intrusions

Components of IDS



Signature-Based IDS

Basic IDS Techniques

WILLIAM & MARY

1. Misuse detection
   - use attack *signatures* (characteristics of real attacks, e.g., illegal sequences of system calls, invalid packets, etc.)
   - can only detect already-known attacks
   - false positive rate is low, but false negative rate is high

Detection (Cont'd)

WILLIAM & MARY

2. Anomaly detection
   - uses a model of "normal" system behavior
   - tries to detect deviations from this behavior, e.g., raises an alarm when a statistically rare event occurs
   - can potentially detect new (not previously-encountered) attacks
   - low false negative rate, high false positive rate
- Which is better?

Ex.: Misuse vs. Anomaly Detection

WILLIAM & MARY

Password file modified	?
Four failed login attempts	?
Failed connection attempts on 50 sequentially-numbered ports	?
User who usually logs in around 10am from dorm logs in at 4:30am from an IP address in Lower Slobovia	?
UDP packet to port 1434 (Slammer Worm)	?

Example Signatures

WILLIAM & MARY

- A sequence of connection attempts to a large number of ports
- A privileged program spawning a shell
- A network packet that has lots of NOOP instruction bytes in it
- Program input containing a very long string (parameter value)
- A large number of TCP SYN packets sent, with no ACKs coming back

Signature Generation

WILLIAM & MARY

- Research challenge: fast, automatic extraction of signatures for new attacks
  - *honeypots* are useful for attracting attacks to generate signatures
- Attack signatures are usually very specific
  - automated engines now generate unlimited variants of a single attack
  - program obfuscation, self-decrypting code
- Possible response: find attack characteristics that are difficult to conceal / obfuscate

Anomaly-Based IDS

Anomaly Detection

WILLIAM & MARY

- Collect a profile of "normal" behavior
  - called *training phase*
  - works best for small, well-defined, stable systems
- IDS compares operational system to this profile, and flags deviations

Examples of Metrics

WILLIAM & MARY

- Count of the number of occurrences of an event per unit time
  - if count exceeded, raise an alarm
- Time elapsed between events
  - if time too small, raise an alarm
- Resource utilization
  - if utilization too high, raise an alarm
- Statistical measures
  - mean, standard deviation, etc.

Examples (Cont'd)

WILLIAM & MARY

- Markov process: use expected likelihood of transition from one system state to another, or from one output to another
- Short sequences of events
  - ex. suppose the normal sequences of system calls during execution of two programs has been measured
  - any serious deviation from such sequences will be flagged as possible signs of an attack

Building Profiles

WILLIAM & MARY

- Profiles are updated regularly, and older data must be "aged" out
  - ex.: $m_t = \alpha$ * most recent measured value + $(1-\alpha)*m_{t-1}$
  - where m_t is expected value for time period t, α is an experimentally-derived weighting factor between .5 and 1.0
- Risk: attacker trains IDS to accept his activity as normal
  - i.e., training data should be free of intrusions, or intrusions must be properly classified in the training data!

Examples of Data Mining Techniques

WILLIAM & MARY

- Association rule learning (find interesting relations between variables)
- Principal components analysis (isolate and focus on the high variance variables)
- Cluster analysis (group data into categories based on similarities)
- …

Conventional View

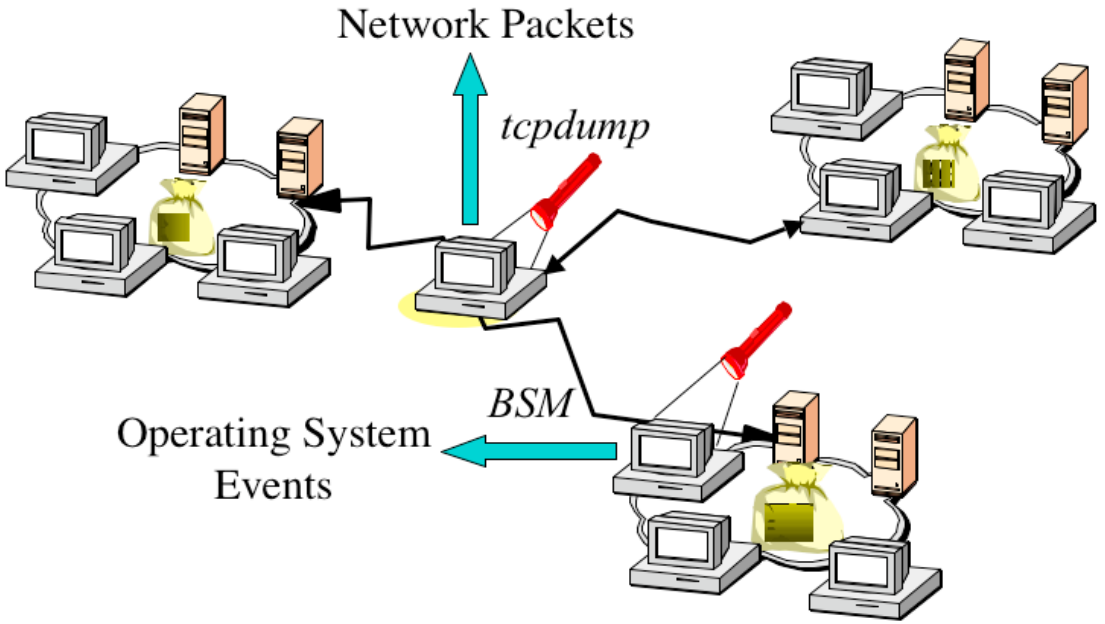
WILLIAM & MARY

- Anomaly-based IDS by itself generates too many false positives
- Combination of anomaly-based and signature-based is best

Host-Based vs. Network-Based IDS

Where Is the IDS Deployed?

- Host-based intrusion detection system (HIDS)
  - monitor activity on a single host
- Network-based intrusion detection (NIDS)
  - monitor traffic, examine packet headers and payloads



Host-Based IDS

WILLIAM & MARY

- Use OS auditing and monitoring mechanisms to find applications taken over by an attacker. Ex.:
  - log all system events (e.g., file accesses)
  - monitor shell commands and system calls executed
- Advantage: better visibility into behavior of individual applications running on the host
- Example application: detecting rootkits

Host-Based (Cont'd)

WILLIAM & MARY

- Drawbacks / limitations
  - need an IDS for every machine
  - if attacker takes over machine, can tamper with IDS binaries and modify audit logs
  - only local view of the attack

Rootkit

WILLIAM & MARY

- Rootkit is a set of "Trojan" system binaries
- Break into a host, download rootkit by FTP, unpack, compile and install
- Possibly turn off anti-virus / IDS
- Hides its own presence!
  - installs hacked binaries for common system monitoring commands, e.g., **`netstat`**, **`ps`**, **`ls`**, **`du`**, **`login`**
- "Sniff" user passwords

File Integrity Checking

WILLIAM & MARY

- Tripwire
  - Records hashes of critical files and binaries
  - System periodically checks that files have not been modified by re-computing and comparing hash
- Ways to bypass?

Network-Based IDS

WILLIAM & MARY

- Inspects network traffic
  - passive (unlike packet-filtering firewalls)
  - often handled by a router or firewall
- Monitors user activities
  - e.g., protocol violations, unusual connection patterns, attack strings in packet payloads
- Advantage: single NIDS can protect many hosts and look for widespread patterns of activity

Network-Based … (cont'd)

WILLIAM & MARY

- Drawbacks / limitations
  - may be easily defeated by encryption (data portions and some header information can be encrypted)
  - not all attacks arrive from the network
  - must monitor, record and process huge amount of traffic on high-speed links
- Attack: overload NIDS with huge data streams, then attempt the intrusion

Popular NIDS : Snort

WILLIAM & MARY

- Popular open-source tool
- Large (> 4000) ruleset for vulnerabilities; Ex.:

"Date: 2005-04-05
Synopsis: the Sourcefire Vulnerability Research Team (VRT) has learned of serious vulnerabilities affecting various implementations of Telnet […] Programming errors in the telnet client code from various vendors may present an attacker with the opportunity to overflow a fixed length buffer […]
Rules to detect attacks against this vulnerability are included in this rule pack"

Some Snort Rule Categories

WILLIAM & MARY

- Backdoors
- Chat
- DDoS
- Finger
- FTP
- ICMP
- IMAP

Multimedia
MySQL
NETBIOS
NNTP
Oracle
P2P

POP
RPC
Scan
Shellcode
SMTP
SNMP
SQL

Telnet
TFTP
Virus
Web…
X11

Snort Rule Syntax

WILLIAM & MARY

- Each snort rule has two logical sections: rule header and rule options
  - rule header contains action, protocol, source (IP address/port), direction, destination (IP address/port)
  - rule option contains alert messages, info on which parts of packet to be inspected

Snort Rule Examples

WILLIAM & MARY

- alert icmp $EXTERNAL_NET any <> $HOME_NET any
  (msg:"DDOS Stacheldraht agent->handler (skillz)";
  content:"skillz";
  itype:0;
  icmp_id:6666; reference:url,staff.washington.edu/dittrich/misc/stacheldraht.analysis;
  classtype:attempted-dos;
  sid:1855; rev:2;)
- alert any any -> 192.168.1.0/24 any
  (flags:A; ack:0; msg: “NMAP TCP ping”;)
  # nmap send TCP ACK pkt with ack field set to 0
- alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS
  (msg:"WEB-IIS cmd.exe access"; flow:to_server,established;
  content:"cmd.exe";
  nocase;
  classtype:web-application-attack;
  sid:1002; rev:5;)

Detecting Attack Strings

WILLIAM & MARY

- Scanning for a signature in each packet is not enough
  - attacker can split attack string into several packets; will defeat stateless NIDS
- Recording just previous packet's text is not enough
  - attacker can send packets out of order
- Attacker can use TCP tricks so that certain packets are seen by NIDS but dropped by the receiving application

Honeypots

WILLIAM & MARY

- Decoy systems to lure attackers
  - away from accessing critical systems
  - to collect information of their activities
  - to encourage attacker to stay on system so administrator can respond
- Filled with fabricated information
- High-interaction decoy vs. Low-interaction decoy
- Instrumented to collect detailed information on attackers activities
- May be single or multiple networked systems

Summary

WILLIAM & MARY

1. Firewalls widely used, packet filters most common
   - one valuable technique among many
2. IDS (both host-based and network-based) widely used
3. Attacks are constantly evolving; the "arms race"
4. False alarm volume, and providing clear feedback to administrators, is a problem