INF3510 Information Security
University of Oslo
Spring 2010

Lecture 8

Perimeter Security

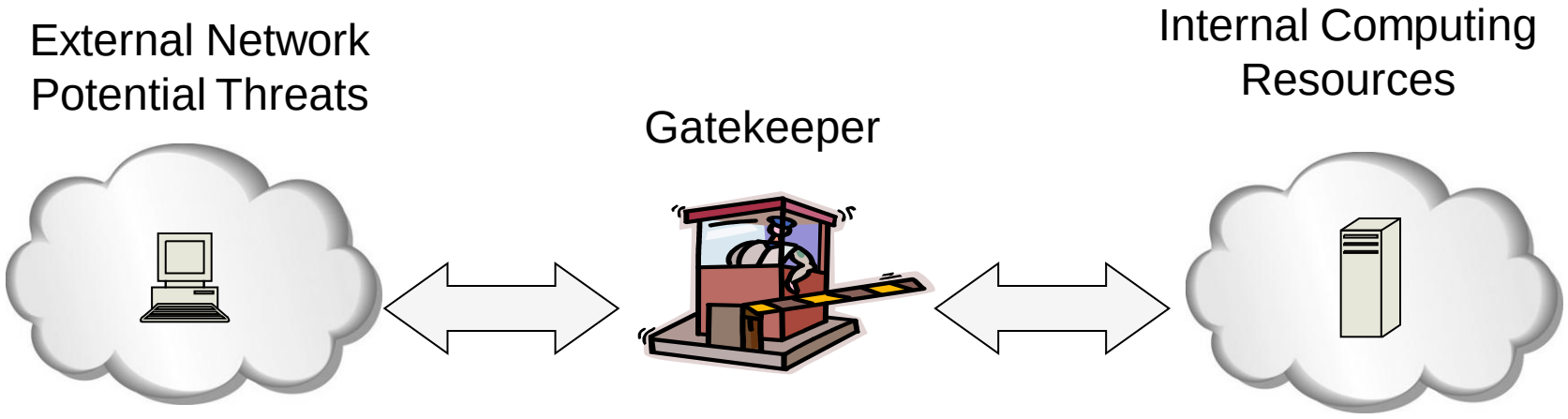
Audun Jøsang

Outline

- Firewalls
  - Routers
  - Proxies
  - Architectures
- Intrusion Detection Systems
  - Host-based
  - Network based
  - Dealing with false alarms

Network Perimeter Security Model: Network Access Control

- A gatekeeper function protects the internal information system against attack from the outside network
- The internal network performs accounting and auditing in order to detect an intrusion



Network Perimeter Security Functions

- Firewalls procedures designed to deny access to all but authorized users. Routers, gateways, proxies
  - The term “firewall” can give false associations and therefore be bad for security usability
  - Can also detect and reject worms, viruses, …
- Intrusion detection: internal controls to monitor and analyze activity in order to detect intruders.
- Honeypots: traps with non-published network services. Access is by definition malicious

Firewalls

See: http://csrc.nist.gov/publications/nistpubs/800-10/main.html

Firewalls: Overview

- A firewall is a system designed to prevent unauthorized access to or from a private network.
- Firewalls can be implemented in both hardware and software, or a combination of both.
- Firewalls are frequently used to prevent unauthorized Internet users from accessing private networks connected to the Internet, especially intranets.
- All messages entering or leaving the intranet pass through the firewall, which examines each message and blocks those that do not meet the specified security criteria.

Firewalls: Overview

- If the level of risk associated with maintaining a connection between an organisation's internal network and the Internet (or some other network(s)) is unacceptable, the most effective way of treating the risk is to avoid the risk altogether and disconnect completely.
- If this is not possible, then firewalls may provide an effective control suitable for reducing the level of risk to an acceptable level.
- A firewall is a component or set of components that restricts access between a protected network and other sets of networks
  - Firewalls are often the first line of defence against attackers but should not be the only defence

Firewalls: Overview

- Firewalls must be effectively administered, updated with the latest patches and monitored.
- Firewalls are only effective if they are set up to implement a well formulated security policy
  - i.e. the most important aspect of a firewall is a clear conception of what it is meant to protect.
- Under the security policy, acceptable traffic must be clearly defined and only that traffic allowed to pass through the firewall; everything else is considered unacceptable and must be stopped by the firewall.

Firewalls:
Types of Firewall Technology

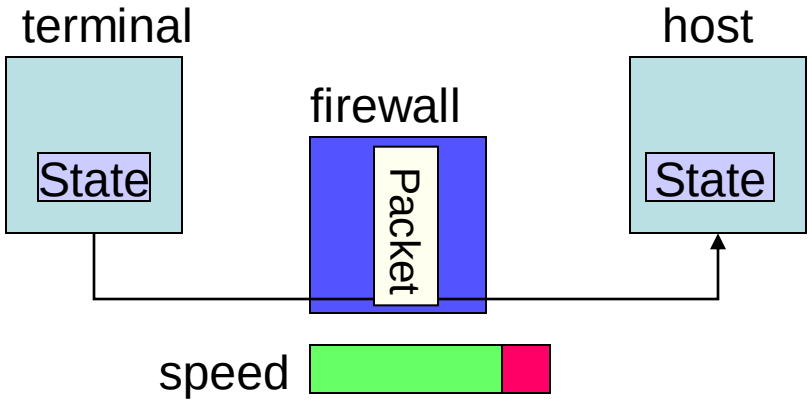
- Simple Packet Filters
- Stateful Packet Filters
- Application Gateways
- Circuit Level Gateways

Firewalls: Router Packet Filter

- A network router function that accepts/rejects packets based on headers is referred to as a packet filter.
- Packet filters examine each packet's headers and make decisions based on attributes such as:
  - Source or Destination IP Addresses
  - Source or Destination Port Numbers
  - Protocol (UDP, TCP or ICMP)
  - ICMP message type
  - And which interface the packet arrived on

Firewalls:
Router Packet Filters

- A packet filter examines each packet that attempts to pass through the filter.
  - Usually this is done for both directions (entering and leaving a network/host)
- Each packet is examined independently of other packets that may be part of the same connection



Router Packet Filtering:
Packet header is inspected
Single packet attacks caught
Very little overhead in firewall
High volume filter

Firewalls: Host-based Packet Filters

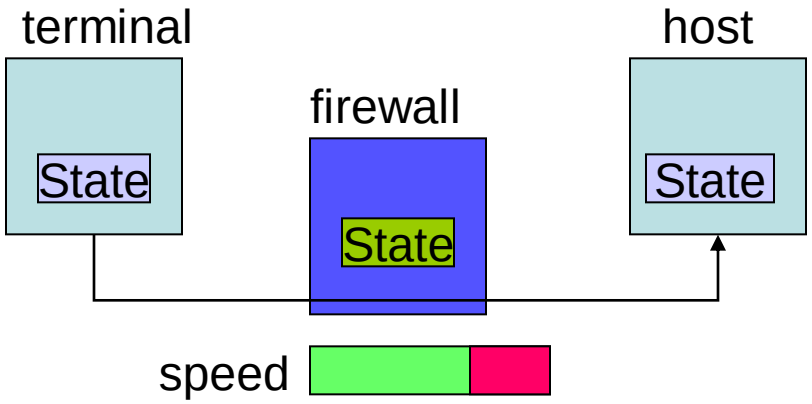
- Routers are commonly used as packet filters, in addition to normal routing duties
- A host can perform packet filtering as well as other duties, such as web serving
  - in this case the packet filter is designed to protect the host itself, not other hosts
- Common packet filter software includes:
  - IPChains for Linux (superseded)
  - TCP Wrappers for various Unix
  - IP Filter for Sun Solaris

Firewalls: Stateful Packet Filters

- Stateful packet filters take account of the current state of a connection
- Stateful packet filters are more ‘intelligent’ than simple packet filters.
- Stateful packet filters are able to recognise if a particular packet is part of an established connection by ‘remembering’ recent traffic history.
- This makes the definition of filtering rules easier to accomplish and therefore potentially more secure.

Stateful packet filter

- A statefull packet filter keeps track of sessions
- Requires memory
- Can be subject to DOS (Denial of Service) attacks



Stateful Inspection
Most multi-packet attacks caught
More fields in packet header inspected
Little overhead in firewall: quick

Firewalls: Stateful Packet Filters

- Sometimes called dynamic packet filters due to their ability to add rules ‘on the fly’. For example:
  - Can recognise an outgoing connection request from an internal client being sent to an external server,
  - And will add a temporary rule to allow the reply traffic back through the firewall.
  - When session is finished, the temporary rule is deleted.
- Common software packages include:
  - IPTables for Linux
  - Checkpoint Firewall-1
  - Cisco PIX (integrated hardware & software)
  - Microsoft Internet Security and Acceleration Server

Firewalls: Packet Filter Strengths and Weaknesses

- Strengths:
  - Low overhead and high throughput
  - Supports almost any application
- Weaknesses:
  - Do not usually interpret application layer data/commands
    - may allow insecure operations to occur
  - Allows direct connection between hosts inside & outside firewall
  - Non-stateful packet filters only: less secure and more difficult to write complex rules

Firewalls: Personal Firewalls

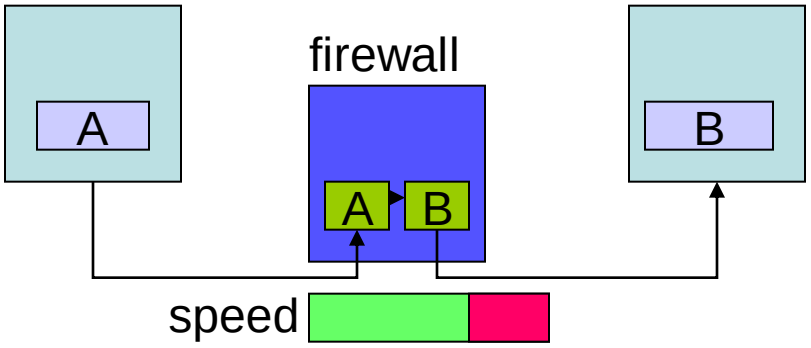
- A personal firewall is a program that is designed to protect the computer on which it is installed.
- Personal firewalls are frequently used by home users to protect themselves from the Internet.
- Personal firewalls are usually a stateful packet filter.
- Some products include anti-virus software as well (usually at extra cost).
  - Vendors such as ZoneAlarm, and Sygate provide a free version of their product for personal use.
  - Windows clients (XP, W7) and Windows servers ship with Internet Connection Firewall (ICF).

Firewalls: Circuit Level Gateways

- A circuit level gateway is a special type of application level gateway with reduced security checking.
- It acts as a relay of TCP/UDP layer data rather than application data, and usually no analysis of the application layer data is performed.
- Connections are validated before allowing data to be exchanged.
- Able to identify a particular packet as being part of a particular connection
  - Limited additional security checking once packet has been identified

Firewalls:
Circuit Level Gateway

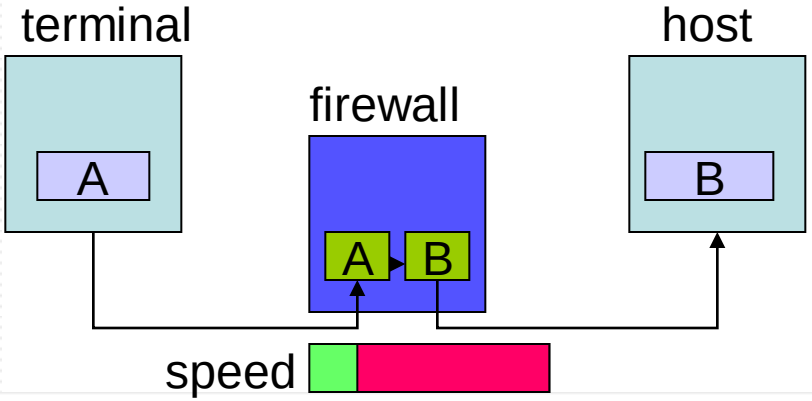
- High performance possible due to limited security checking
- Similar strengths and weaknesses to stateful packet filters except
  - Can examine application layer data to a certain extent, but not up to application level gateway standards
  - E.g: Some control/blocking of insecure FTP commands



Circuit-Level Firewall:
Packet session terminated and recreated via a Proxy Server
All multi-packet attacks caught
Packet header completely inspected
High overhead in firewall: slow

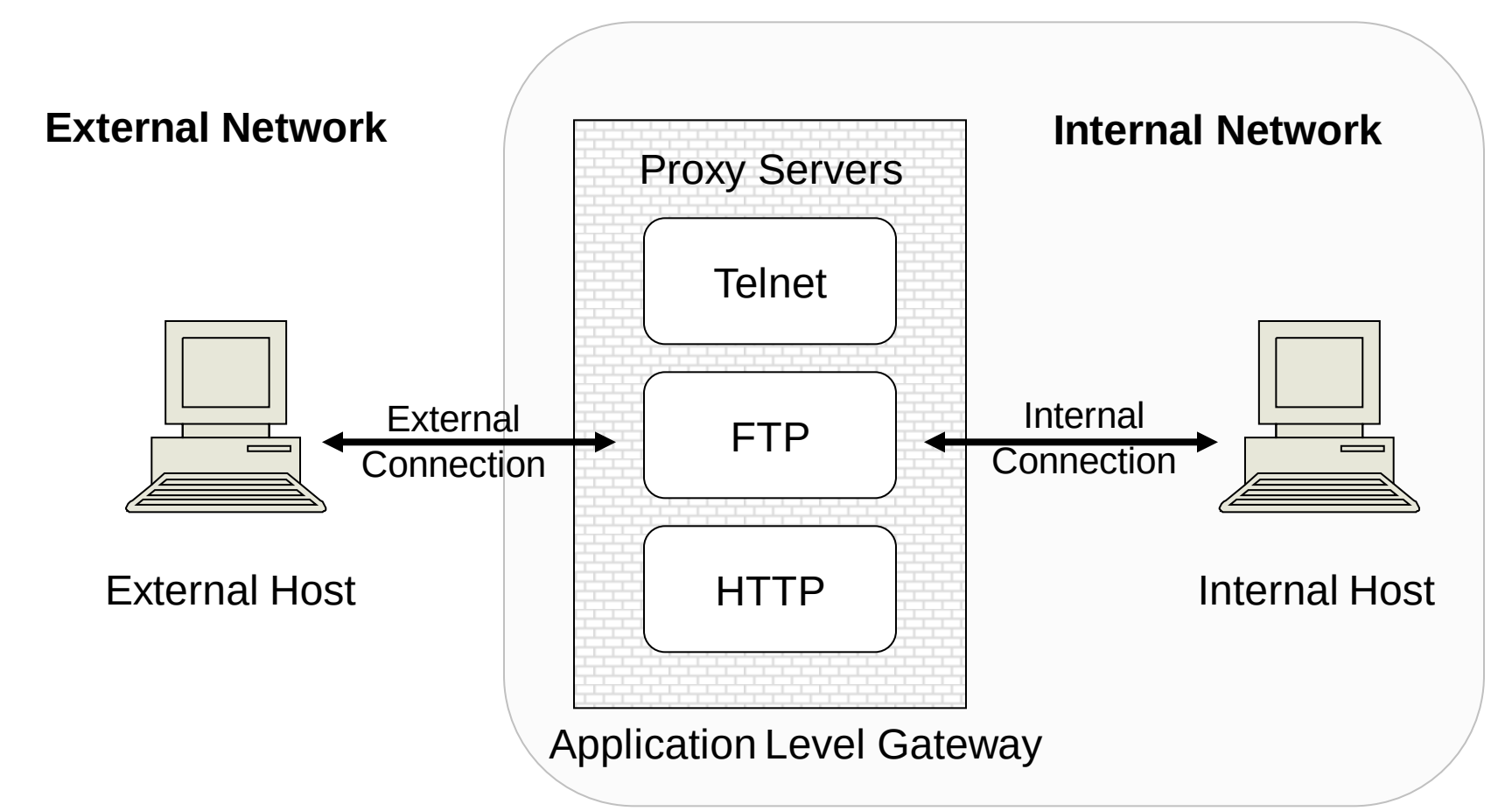
Firewalls: Application Level Gateway

- Acts as a relay of application level traffic
- Also known as an application proxy because the firewall needs to act on behalf of the client
- Usually configured to support only specific applications or specific features of an application
  - each application supported by a specific gateway in the firewall



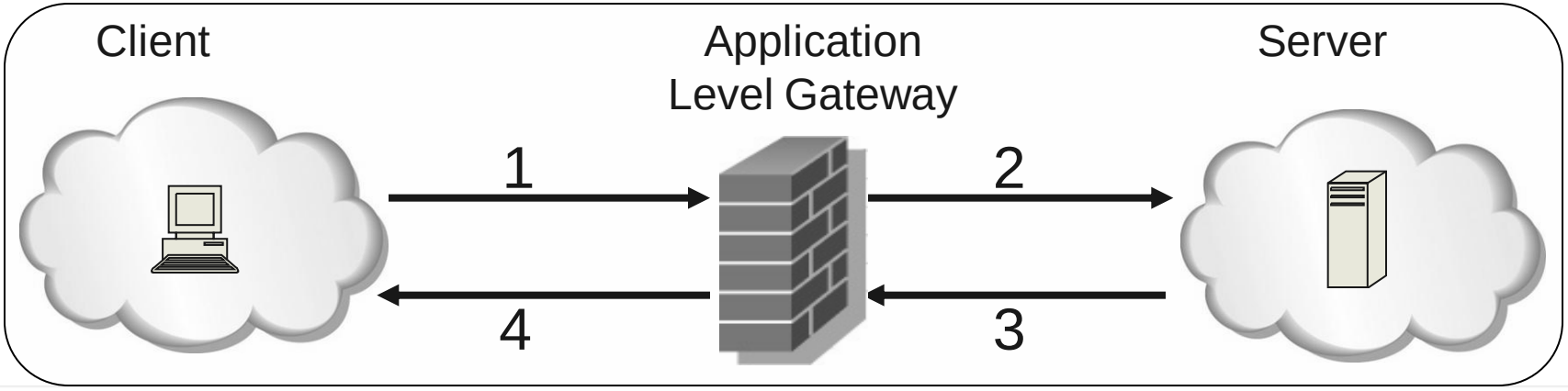
Application-Level Firewall
Packet session terminated and recreated via a Proxy Server
Packet header completely inspected
Most or all of application inspected
Highest overhead: slow & low volume

Firewalls: Application Level Gateway



Firewalls: Application Level Gateway

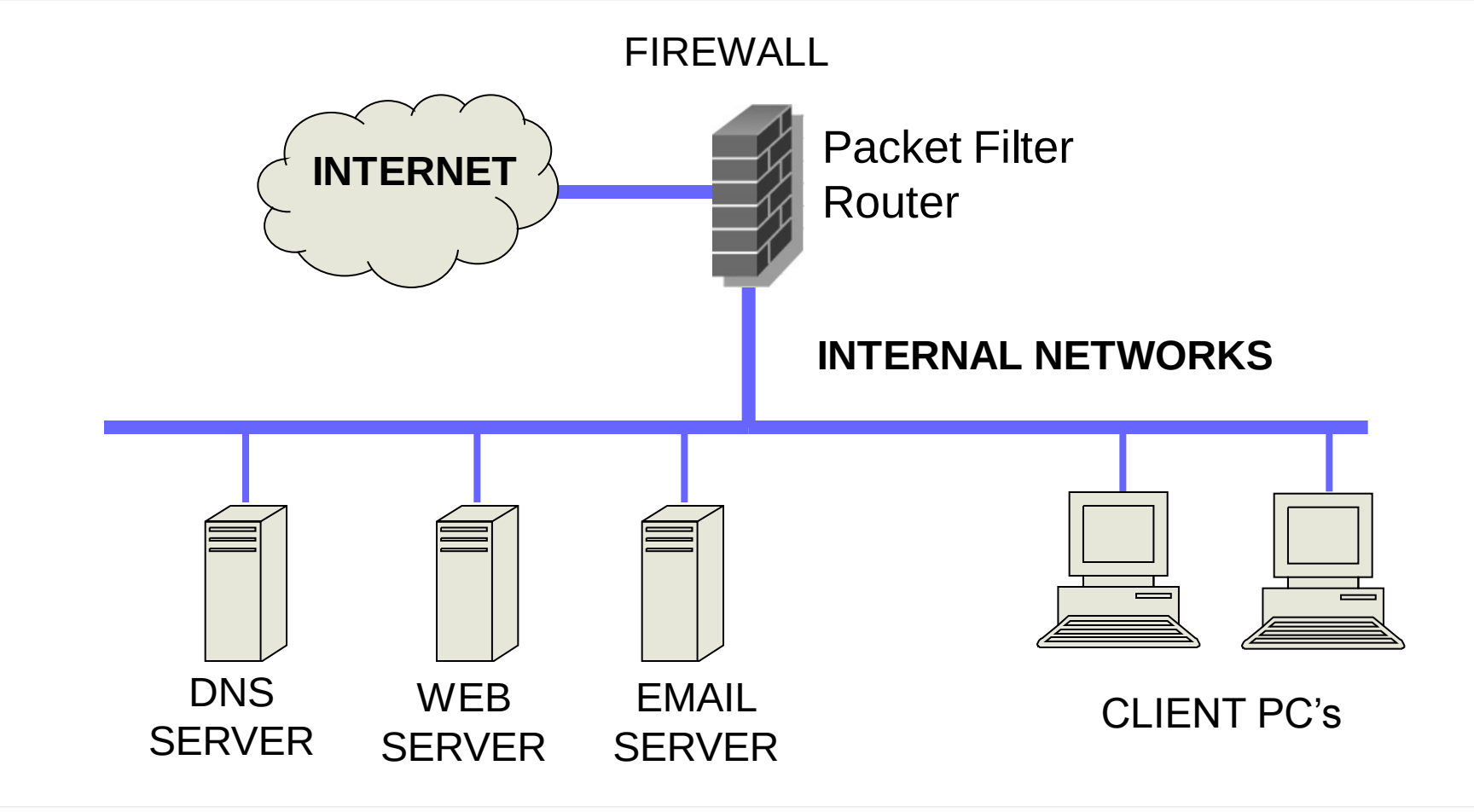
- Client sends a request to the server, which is intercepted by the firewall (application gateway)
- Firewall sends the request to the server on behalf of client.
- Server sends reply back to the firewall.
- Firewall sends reply to the client.
- Both client and server think they are communicating with each other, not knowing the firewall exists. It is **transparent**.



Firewalls: Application Gateway

- Strengths:
  - Easy logging and audit of all incoming traffic
  - Provides potential for best security through control of application layer data/commands
- Weaknesses:
  - May require some time for vendor to write new gateways for new applications
  - Requires one more additional connection (including processing resources) for each new connection
  - Slower than packet filters

Firewalls:
Simple Firewall Architecture



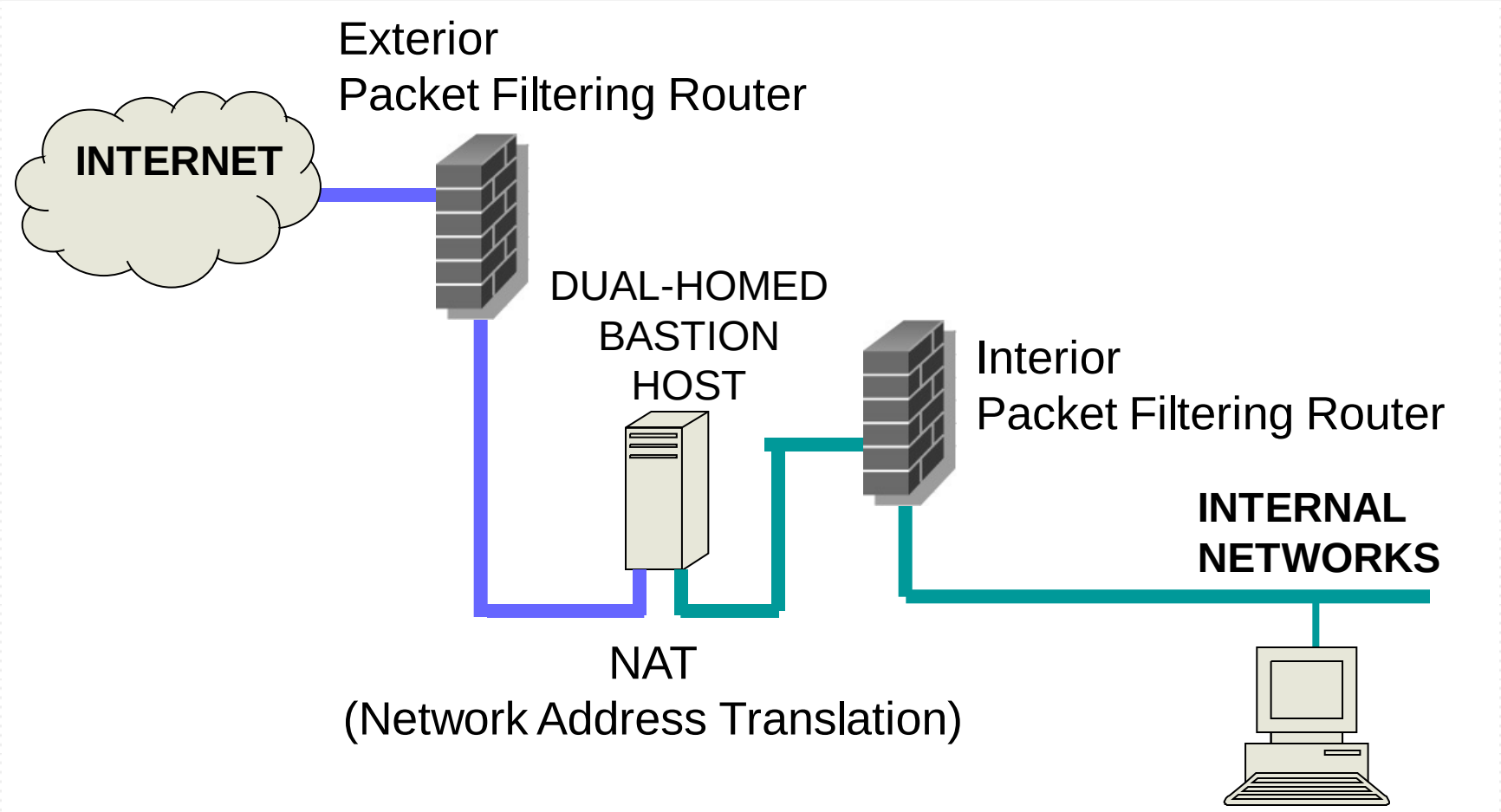
Network Address Translation (NAT)

- Translates network addresses & ports
- Possibilities:
  - One external address per internal address
  - Dynamically assign external address
  - Map multiple internal to one external (port sharing)
  - Dynamically assign external addresses and ports
- Use non-routable address ranges:
  - 10.0.0.0-10.255.255.255,
  - 172.16.0.0-172.31.255.255 or
  - 192.168.0.0-192.168.255.255 for local networks

Network Address Translation (NAT)

- Advantages
  - Helps enforce control over outbound connections
  - Helps restrict incoming traffic
  - Helps conceal internal network configuration
  - Prevents port scanning
- Can’t be used with:
  - protocols that require a separate back-channel
  - protocols that encrypt TCP headers
  - embed TCP address info
  - specifically use original IP for some security reason

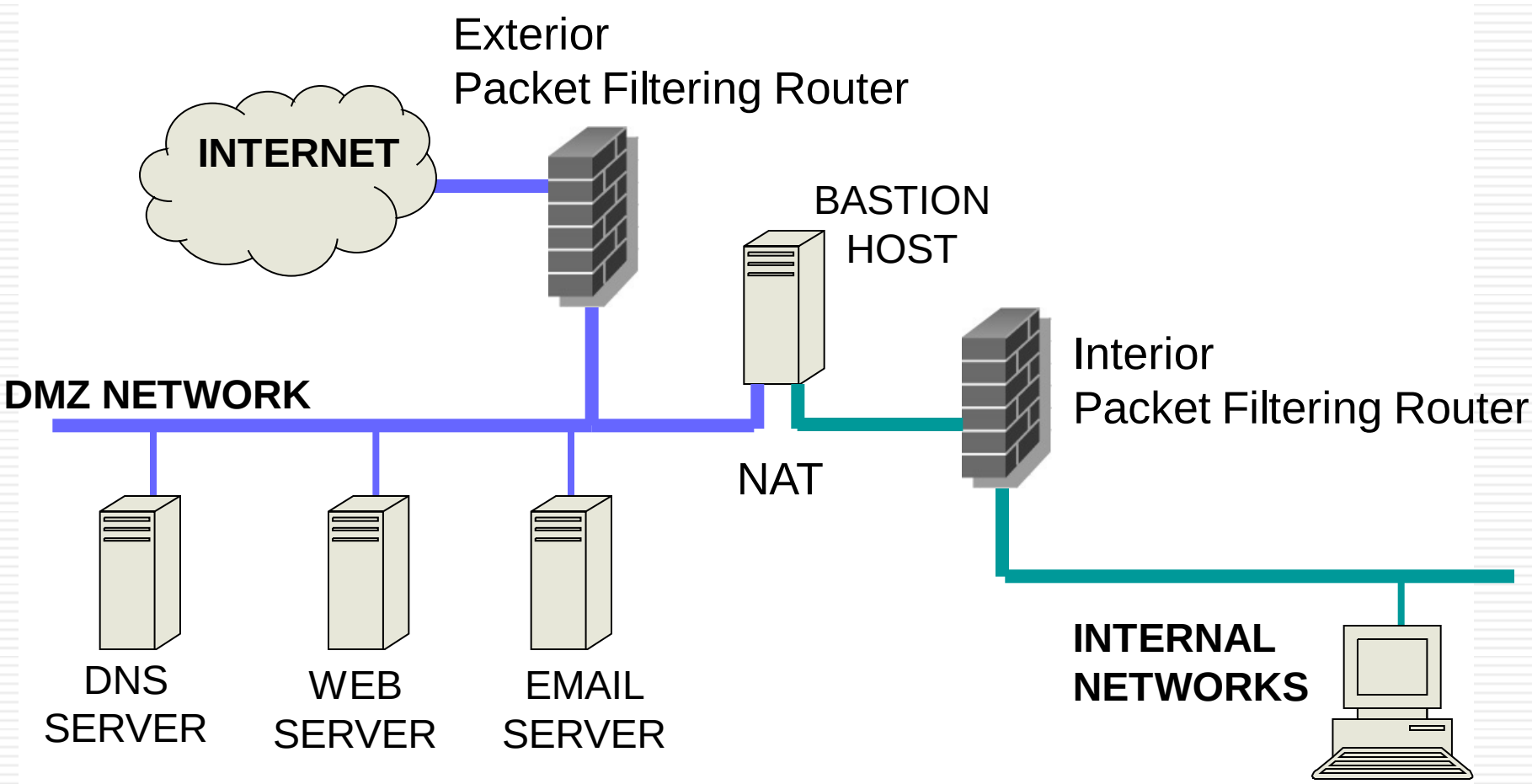
Dual-Homed Screened Bastion Host and Network Address Translation



Screened Bastion-Host

- The Screened Bastion-Host is a dedicated firewall that comes in addition to the packet filtering routers
- Functions
  - Proxy for services in the internal network
  - Dual-homed, i.e. with separate NICs (Network Interface Cards)
  - Network Address Translation (NAT)
  - Protocol gateway for different link layer protocols

Screened Subnet DMZ Architecture



Screened Subnet
Defence in layers

- An exterior packet filtering router is the first line of defence, reduces traffic
- The DNS, Email and Web servers provide services to the Internet and are therefore at some risk of attack. These servers are on a network segment that is neither completely outside nor completely inside the organisation, called the DMZ (DeMilitarized Zone)
- NAT and proxy services provide additional defences.
- The interior router usually also controls what traffic can leave the internal networks bound for external destinations, another point of enforcing security policy.

Intrusion Detection Systems

Intrusion Detection Systems: Overview

- Intrusion detection systems (IDS) are automated systems (programs) that detect suspicious events
- IDS can be either host-based or network-based.
- A host based IDS is designed to detect intrusions only on the host it is installed on
  - monitor changes to host's operating system files and traffic sent to the host
- Network based IDS are designed to detect intrusions on one or more network segments, usually deployed to protect a number of hosts
  - monitor network/s looking for suspicious traffic

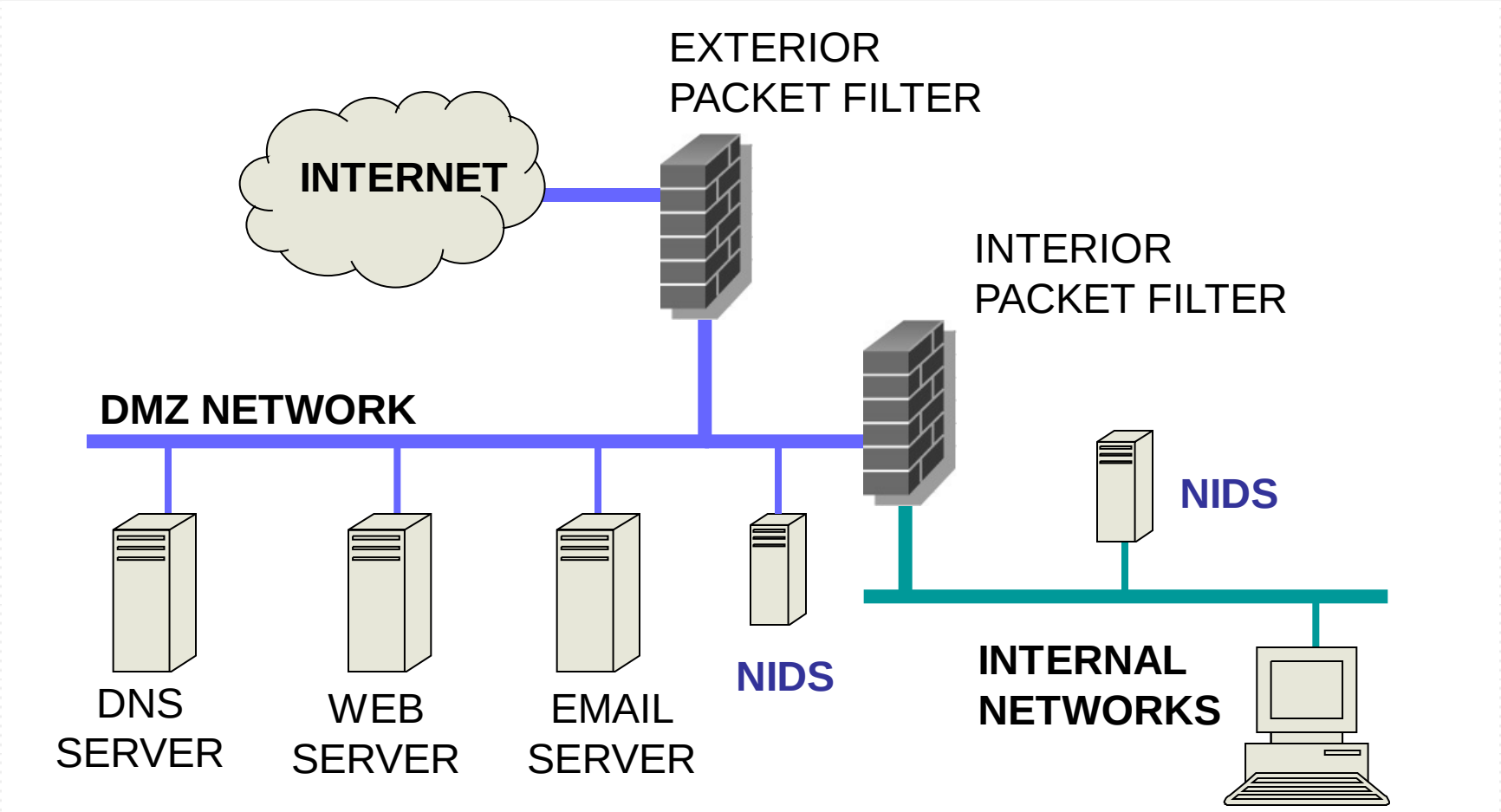
What Should Be Detected?

- Attempted and successful break-ins
- Attacks by legitimate users
  - For example, illegitimate use of root privileges
  - Unauthorized access to resources and data
- Trojan horses
- Viruses and worms
- Denial of service attacks

Where Are IDS Deployed?

- Host-based
  - Monitor activity on a single host
  - Advantage: better visibility into behavior of individual applications running on the host
- Network-based (NIDS)
  - Often placed on a router or firewall
  - Monitor traffic, examine packet headers and payloads
  - Advantage: single NIDS can protect many hosts and look for global patterns

IDS:
Network IDS Deployment



Intrusion Detection Techniques

- Misuse detection
  - Use attack "signatures" (need a model of the attack)
    - Sequences of system calls, patterns of network traffic, etc.
  - Must know in advance what attacker will do (how?)
  - Can only detect known attacks
- Anomaly detection
  - Using a model of normal system behavior, try to detect deviations and abnormalities
    - E.g., raise an alarm when a statistically rare event(s) occurs
  - Can potentially detect unknown attacks
- Which is harder to do?

Misuse vs. Anomaly

• Password file modified	Misuse
◆ Four failed login attempts	Anomaly
◆ Failed connection attempts on 50 sequential ports	Anomaly
◆ User who usually logs in around 10am from a UT dorm logs in at 4:30am from a Russian IP address	Anomaly
◆ UDP packet to port 1434	Misuse
◆ “DEBUG” in the body of an SMTP message	Not an attack! (most likely)

Misuse Detection (Signature-Based)

- Set of rules defining a behavioral signature likely to be associated with attack of a certain type
  - Example: buffer overflow
    - A setuid program spawns a shell with certain arguments
    - A network packet has lots of NOPs in it
    - Very long argument to a string function
  - Example: SYN flooding (denial of service)
    - Large number of SYN packets without ACKs coming back
    - …or is this simply a poor network connection?
- Attack signatures are usually very specific and may miss variants of known attacks
  - Why not make signatures more general?

Extracting Misuse Signatures

- Use invariant characteristics of known attacks
  - Bodies of known viruses and worms, port numbers of applications with known buffer overflows, RET addresses of overflow exploits
  - Hard to handle mutations
    - Polymorphic viruses: each copy has a different body
- Big research challenge: fast, automatic extraction of signatures of new attacks
- Honeypots are useful for signature extraction
  - Try to attract malicious activity, be an early target

Anomaly Detection

- Define a profile describing “normal” behavior
  - Works best for “small”, well-defined systems (single program rather than huge multi-user OS)
- Profile may be statistical
  - Build it manually (this is hard)
  - Use machine learning and data mining techniques
    - Log system activities for a while, then “train” IDS to recognize normal and abnormal patterns
  - Risk: attacker trains IDS to accept his activity as normal
    - Daily low-volume port scan may train IDS to accept port scans
- IDS flags deviations from the “normal” profile

What's a "Profile?"

- Login and session activity
  - Login and location frequency; last login; password fails; session elapsed time; session output, CPU, I/O
- Command and program execution
  - Execution frequency; program CPU, I/O, other resources (watch for exhaustion); denied executions
- File access activity
  - Read/write/create/delete frequency; records read/written; failed reads, writes, creates, deletes; resource exhaustion
- How to make all this auditing scalable?

IDS:
Major IDS Operational Issue

- A major problem with IDS in general is the number of false positive alarms they generate
  - Can lead to a sense of mistrust, then apathy by security administrator
- Despite this, IDS are becoming more popular
- Common products include:
  - ISS RealSecure
  - Snort (open source NIDS)

Host-Based IDS

- Use OS auditing and monitoring mechanisms to find applications taken over by attacker
  - Log all relevant system events (e.g., file accesses)
  - Monitor shell commands and system calls executed by user applications and system programs
    - Pay a price in performance if every system call is filtered
- Con: need an IDS for every machine
- Con: if attacker takes over machine, can tamper with IDS binaries and modify audit logs
- Con: only local view of the attack

Level of Monitoring

- Which types of events to monitor?
  - OS system calls
  - Command line
  - Network data (e.g., from routers and firewalls)
  - Processes
  - Keystrokes
  - File and device accesses

Tripwire

- File integrity checker
  - Records hashes of critical files and binaries
    - Recorded hashes must be in read-only memory (why?)
  - Periodically checks that files have not been modified, verifies sizes, dates, permission
- Good for detecting rootkits
- Can be subverted by a clever rootkit
  - Install backdoor inside a continuously running system process (no changes on disk!)
  - Copy old files back into place before Tripwire runs
- How to detect modifications to running process?

Network IDS Deployment

- A Network IDS (NIDS) needs to monitor traffic for the network/s it is protecting
- Can be used to enhance the security provided by a firewall by adding another layer of defence
- Generally accomplished by placing the network interface card of the IDS in ‘promiscuous’ mode to capture all network traffic that crosses its network segment (not just traffic destined for the IDS itself)
- Multiple IDS can be deployed on multiple networks and report to a central system

Network-Based IDS

- Inspect network traffic
  - For example, use tcpdump to sniff packets on a router
  - Passive (unlike firewalls)
  - Default action: let traffic pass (unlike firewalls)
- Watch for protocol violations, unusual connection patterns, attack strings in packet payloads
  - Check packets against rule sets
- Problems:
  - can’t inspect encrypted traffic (IPsec, VPNs)’
  - not all attacks arrive from the network
  - record and process huge amount of traffic

Popular NIDS

- Snort (popular open-source tool)
  - Large rule sets for known vulnerabilities
    - **2009-03-31:** A programming error in MySQL Server may allow a remote attacker to cause a Denial of Service (DoS) against a vulnerable machine.
    - **2009-03-27:** Microsoft Windows GDI Buffer Overflow: A programming error in the Microsoft Windows kernel may allow a remote attacker to execute code with system level privileges. This may be exploited when specially crafted EMF files are viewed using Microsoft Internet Explorer.
- Bro (developed by Vern Paxson)

  - Separates data collection and security decisions
    - Event Engine distills the packet stream into high-level events describing what's happening on the network
    - Policy Script Interpeter uses a script defining the network's security policy to decide what to do in response

Irony and NIDS

Sourcefire Snort Remote Buffer Overflow

- Notification Type: IBM Internet Security Systems Protection Advisory
- Notification Date: Feb 19, 2007
- Description: Snort IDS and Sourcefire Intrusion Sensor IDS/IPS are vulnerable to a stack-based buffer overflow, which can result in remote code execution.

… patched since then (phew!)

Port Scanning

- Many vulnerabilities are OS-specific
  - Bugs in specific implementations, default configuration
- Port scan is often a prelude to an attack
  - Attacker tries many ports on many IP addresses
    - For example, looking for an old version of some daemon with an unpatched buffer overflow
  - If characteristic behavior detected, mount attack
  - "The Art of Intrusion": virtually every attack involves port scanning and password cracking

Scanning Defense

- Scan suppression: block traffic from addresses that previously produced too many failed connection attempts
  - Goal: detect port scans from attacker-controlled hosts
  - Requires network filtering and maintaining state
  - Can be subverted by slow scanning; does not work very well if the origin of scan is far away (why?)
- False positives are common, too
  - Website load balancers, stale IP caches
    - E.g., dynamically get an IP address that was used by P2P host

Attacking and Evading NIDS

- Overload NIDS with huge data streams, then attempt the intrusion
  - Bro solution: watchdog timer
    - Check that all packets are processed by Bro within T seconds; if not, terminate Bro, use tcpdump to log all subsequent traffic
- Use encryption to hide packet contents
- Split malicious data into multiple packets
  - NIDS does not have full TCP state and does not always understand every command of receiving application
  - Simple example: send “ROB<DEL><BS><BS>OT”, receiving application may reassemble to “ROOT”

Intrusion Detection Problems

- Lack of training data with real attacks
  - But lots of “normal” network traffic, system call data
- Data drift
  - Statistical methods detect changes in behavior
  - Attacker can attack gradually and incrementally
- Main characteristics not well understood
  - By many measures, attack may be within bounds of “normal” range of activities
- False identifications are very costly
  - Sysadm will spend many hours examining evidence

Intrusion Detection Errors

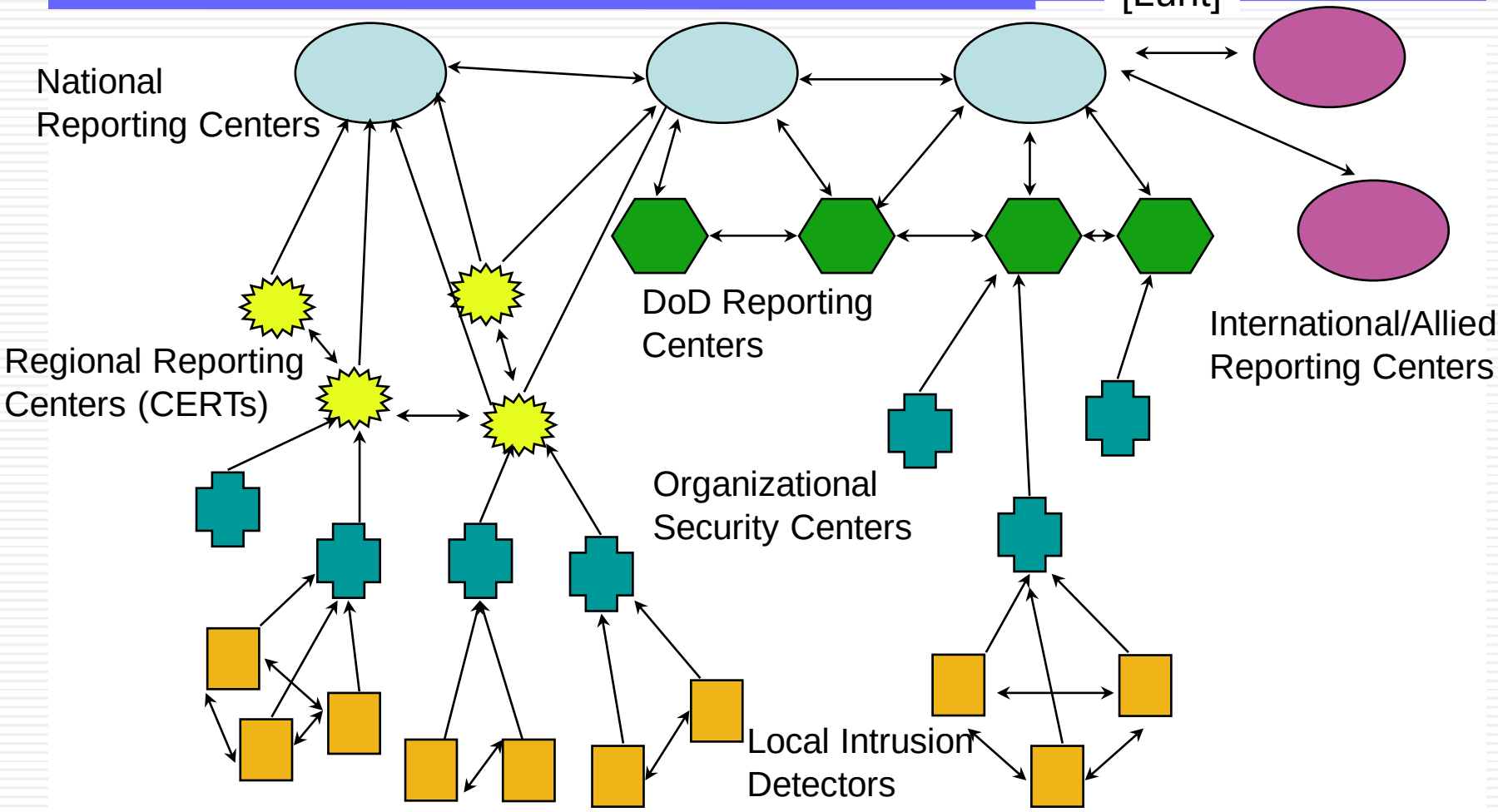
- False negatives: attack is not detected
  - Big problem in signature-based misuse detection
- False positives: harmless behavior is classified as an attack
  - Big problem in statistical anomaly detection
- Both types of IDS suffer from both error types
- Which is a bigger problem?
  - Attacks are fairly rare events
  - IDS often suffer from base-rate fallacy

Base Rate Fallacy

- Consider two statements: r: “attack occurs”
  s: “signature detected”
- Consider the conditional probabilities:
  P(s|r): probability of detecting signature when the attack occurs
  P(r|s): probability of attack when the signature is detected
- Learning new attack patterns produces values for P(s|r) whereas intrusion detection requires knowledge of P(r|s)
- Deriving P(r|s) from P(s|r) requires the base rate of r
  - Hard to determine base rate of r
- Assuming P(r|s) = P(s|r) is an approximation, but can lead to completely false conclusion (base rate fallacy)

Strategic Intrusion Assessment

[Lunt]



Strategic Intrusion Assessment

- Test over two-week period by Air Force Information Warfare Center
  - Intrusion detectors at 100 Air Force bases alarmed on 2,000,000 sessions
  - Manual review identified 12,000 suspicious events
  - Further manual review => four actual incidents
- Conclusion
  - Most alarms are false positives
  - Most true positives are trivial incidents which can be ignored, i.e. the attacks will never be able to penetrate any system
  - Of the significant incidents, most are isolated attacks to be dealt with locally

Network Telescopes and Honeypots

- Monitor a cross-section of Internet address space
  - Especially useful if includes unused “dark space”
- Attacks in far corners of the Internet may produce traffic directed at your addresses
  - “Backscatter”: responses of DoS victims to randomly spoofed IP addresses
  - Random scanning by worms
- Can combine with “honeypots”
  - Any connection to a “honeypot” behind an otherwise unused IP address means an attack (why?)
  - Can use this to extract worm signatures (how?)

Intrusion Prevention Systems

- Intrusion Prevention System (IPS) is a relatively new term that can mean different things
- Most commonly, an IPS is a combination of an IDS and a firewall
- A system that detects an attack and can stop it as well
- Can be application specific
  - Deployed on a host to stop attacks on specific applications such as IIS
- Can be an extension of an NIDS
- Can be used to defend systems that cannot be patched

End of Lecture

We have discussed:

- Firewall techniques
- Intrusion detection techniques