Introduction to Computer Security

Chapter 8: Intrusion Detection

Chi-Yu Li (2019 Spring)
Computer Science Department
National Chiao Tung University

Outline

- Intruders
- Intrusion detection
- Analysis approaches
- Host-based intrusion detection
- Network-based intrusion detection
- Distributed or hybrid intrusion detection
- Honeypots

Intruders

- Cyber criminals
  - Individuals or members of an organized crime group with a goal of financial reward
  - Activities: identity theft, theft of financial credentials, data theft, data ransoming
  - Meet to trade tips in underground forums (e.g., DarkMarket.org, theftservices.com)
- Activists
  - Individuals or members of a larger group of outsider attackers
    - Skill level is often low
  - Goals: promote and publicize their social or political causes
  - Activities: website defacement, DoS attacks, theft and distribution of data

Intruders (Cont.)

- State-sponsored organizations
  - Groups of hackers sponsored by governments to conduct espionage or sabotage activities
  - Known as APTs
  - Widespread nature: from China to the USA, UK, and their intelligence allies
- Others
  - Hackers with motivations other than the above
  - Classic hackers or crackers: motivated by technical challenge or by peer-group esteem and reputation

Three Skill Levels of Intruders

- Apprentice
  - Hackers with minimal technical skill who primarily uses existing attack toolkits
  - Known as “script-kiddies”
  - The easiest to defend against
- Journeyman
  - Hackers with sufficient technical skills to modify/extend attack toolkits
  - Be able to use new vulnerabilities; may be able to locate new ones
  - The changes in attack tools make identifying and defending harder
- Master
  - Hackers with high-level technical skills to discover new vulnerabilities
  - Highest difficulty for defense

Intruder Behavior

- Target acquisition and information gathering
  - Identifying and characterizing the target systems using publicly information
  - Using network exploration tools to map target resources
  - Examples
    - Exploring corporate website for information on structure, personnel, key systems
    - Gathering information on target network using DNS lookup tools (e.g., dig, host)
- Initial access
  - Exploiting a remote network vulnerability, guessing weak authentication credentials, or installing malware
  - Examples
    - Brute force to guess passwords
    - Exploiting vulnerability in Web server to gain system access
    - Sending spear-phishing e-mail to key people

Intruder Behavior (Cont.)

- Privilege escalation
  - Increasing the privileges via a local access vulnerability
  - Examples
    - Exploiting any vulnerable app to gain elevated privileges
    - Installing sniffers to capture administrator passwords
- Information gathering or system exploit
  - Accessing or modifying information or resources on the system
  - Examples
    - Scanning files for desired information

Intruder Behavior (Cont.)

- Maintaining access
  - Installing backdoors or other malicious software
  - Enabling continued access after the initial attack
  - Examples
    - Installing rootkit with backdoor for later access
    - Modifying or disabling anti-virus programs running on system
- Covering tracks
  - Disabling or editing audit logs to remove evidence of attack activity
  - Examples
    - Using rootkit to hide files installed on system

Intrusion Detection

- Definitions from Internet Security Glossary (RFC 2828)
  - Security intrusion: unauthorized act of bypassing the security mechanisms of a system
  - Intrusion detection: a hardware or software function that gathers and analyzes information from various areas within a computer or a network to identify possible security intrusions

Intrusion Detection System (IDS)

- Host-based IDS (HIDS)
  - Monitoring the characteristics of a single host and its events
- Network-based IDS (NIDS)
  - Monitoring network traffic and analyzing network, transport, and app protocols
- Distributed or hybrid IDS
  - Combining information from sensors, often both host and network-based, in a central analyzer

Three logical components

- **Sensors - collect data**
- **Analyzers - determine if intrusion has occurred**
- **User interface - view output or control system behavior**

Example: The Zeek Network Security Monitor

The Zeek Network Security Monitor

Why Choose Zeek? Zeek is a powerful network analysis framework that is much different from the typical IDS you may know. (Zeek is the new name for the long-established Bro system. Note that parts of the system retain the "Bro" name, and it also often appears in the documentation and distributions.)

Adaptable
Zeek's domain-specific scripting language enables site-specific monitoring policies.

Efficient
Zeek targets high-performance networks and is used operationally at a variety of large sites.

Flexible
Zeek is not restricted to any particular detection approach and does not rely on traditional signatures.

Forensics

In-depth Analysis
Zeek comes with analyzers for many protocols, enabling high-level semantic analysis at the application layer.

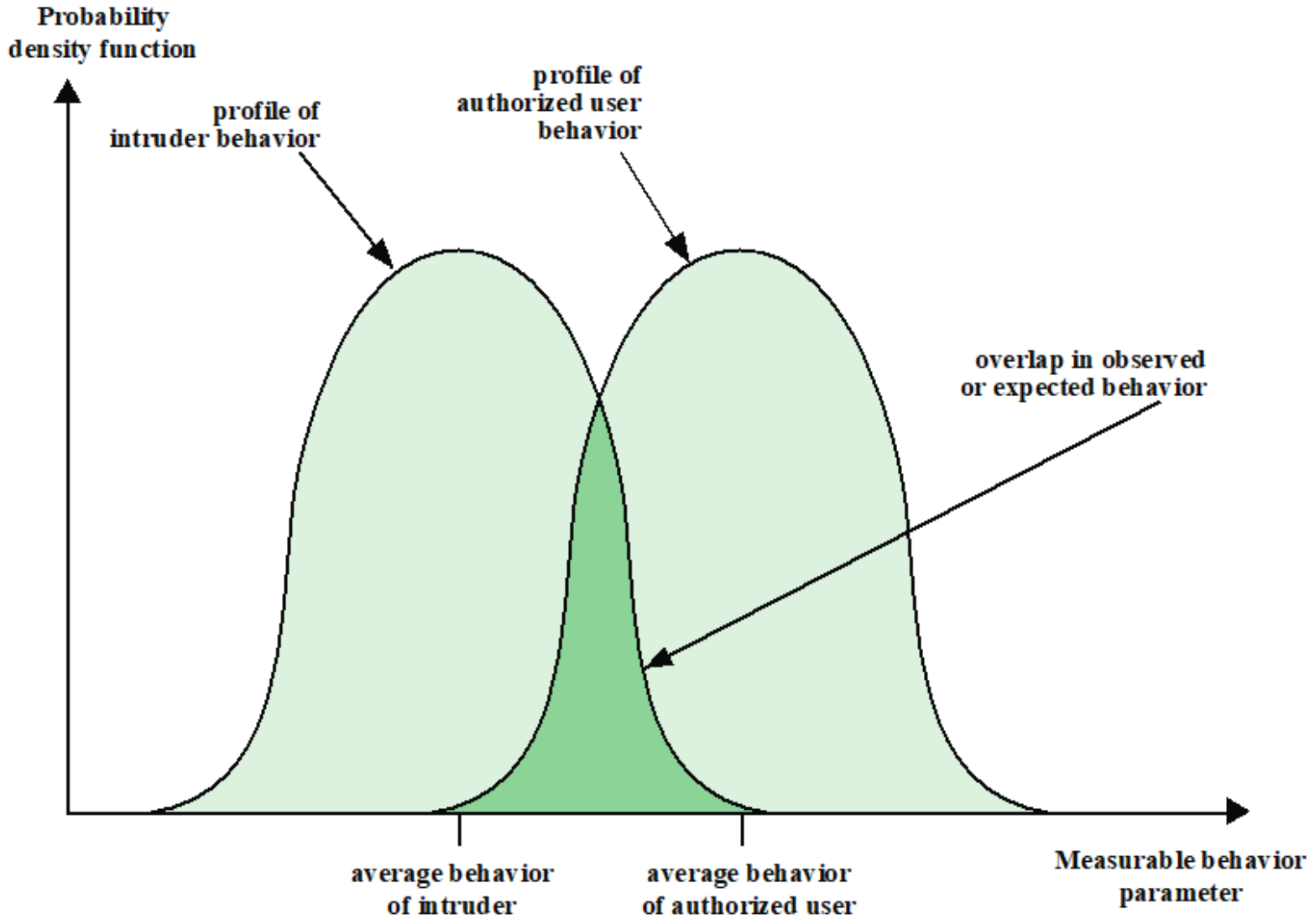
Highly Stateful
Zeek keeps extensive application-layer state about the network it monitors.

Open Interfaces
Zeek interfaces with other applications for real-time exchange of information.

Open Source

Intrusion Detection: Basic Principles

- Another line of defense against intrusions
  - Others: authentication, access control, and firewall
- Based on the assumption: the behavior of the intruder differs from that of a legitimate user



Requirements

Run continually	**Be fault tolerant**	**Resist subversion**
Impose a minimal overhead on system	**Configured according to system security policies**	**Adapt to changes in systems and users**
Scale to monitor large numbers of systems	**Provide graceful degradation of service**	**Allow dynamic reconfiguration**

Analysis Approaches

Anomaly detection

- Collecting data
  - The behavior of legitimate users over a period of time
- Analyzing current observed behavior with a high level of confidence
  - A legitimate user or an intruder

Signature/Heuristic detection

- Using a set of known malicious data patterns (signatures) or attack rules (heuristics)
- Can only identify known attacks

Anomaly Detection: Categories

Statistical	Knowledge based	Machine-learning
• Analysis of the observed behaviors/metrics • Using univariate, multivariate, or time-series models • Pros: **simplicity**, **low computation cost**, and **lack of assumptions about behavior expected** • Cons: **difficulty in selecting suitable mtrics**, and **not all behaviors can be modeled**	• Classifying the observed data using a set of rules • Rules are developed during the training phase, usually manually • Formal tools: finite-state machine or standard description language • Pros: **robustness** and **flexibility** • Cons: **difficulty/time required to develop high-quality knowledge rules**	• Automatically determining a suitable classification model from the training data using data mining techniques • Pros: **flexibility**, **adaptability**, and **ability to capture interdependencies between factors** • Cons: **requiring significant time and computational resources**

Anomaly Detection (Cont.)

- A variety of machine-learning approaches with varying success
  - Bayesian networks, Markov models, neural networks, fuzzy logic, genetic algorithms, clustering and outlier detection

- What is the key limitation?

Signature or Heuristic Detection

- Signature approaches
  - Matching a large collection of known patterns of malicious data in a system or over a network
  - Signatures
    - Large enough to minimize the false alarm rate
    - Sill detecting a sufficiently large fraction of malicious data
  - Widely used in anti-virus products
- Rule-based heuristic identification
  - Identifying known penetrations
  - Identifying suspicious behavior within the bounds of established patterns of usage
  - Specific to the machine and OS

Host-based Intrusion Detection (HIDS)

- A specialized layer of security software to vulnerable or sensitive systems
  - Monitor activity on the system to detect suspicious behavior
- Main purpose: detect intrusions, log suspicious events, and send alerts
  - Can use either anomaly or signature and heuristics approaches
- Primary benefit: can detect both external and internal intrusions

Data Sources and Sensors

- System call traces
  - A record of the sequence of system calls by processes on a system
  - Work well for Unix and Linux systems, but problematic on Windows
  - 95-99% detection rates [CREE13]

- Audit (log file) records
  - Most modern OSes have accounting software that collects information on user activity
  - Pros: no additional collection software is needed
  - Cons: may not contain the needed information or in a convenient form
  - 80% detection rates

Data Sources and Sensors (Cont.)

- File integrity checksums
  - Periodically scan critical files for changes
  - Cons: generate and protect the checksums, difficult to monitor changing files
- Registry access
  - Used on Windows to monitor access to the registry

Anomaly HIDS

Gathering the system call traces using an OS hook, e.g., BSM (Basic Security Module) audit module

Using traces of key DDL function calls

- DDLs (Dynamic Link Libraries): an intermediary between process requests and system call interface

(a) Ubuntu Linux System Calls

accept, access, acct, adjtime, aiocancel, aioread, aiowait, aiowrite, alarm, async_daemon, auditsys, bind, chdir, chmod, chown, chroot, close, connect, creat, dup, dup2, execv, execve, exit, exportfs, fchdir, fchmod, fchown, fchroot, fcntl, flock, fork, fpathconf, fstat, fstat, fstatfs, fsync, ftime, ftruncate, getdents, getdirentries, getdomainname, getdopt, getdtablesize, getfh, getgid, getgroups, gethostid, gethostname, getitimer, getmsg, getpagesize, getpeername, getpgrp, getpid, getpriority, getrlimit, getrusage, getsockname, getsockopt, gettimeofday, getuid, gtty, ioctl, kill, killpg, link, listen, lseek, lstat, madvise, mctl, mincore, mkdir, mknod, mmap, mount, mount, mprotect, mpxchan, msgsys, msync, munmap, nfs_mount, nfssvc, nice, open, pathconf, pause, pcfs_mount, phys, pipe, poll, profil, ptrace, putmsg, quota, quotactl, read, readlink, readv, reboot, recv, recvfrom, recvmsg, rename, resuba, rfssys, rmdir, sbreak, sbrk, select, semsys, send, sendmsg, sendto, setdomainname, setdopt, setgid, setgroups, sethostid, sethostname, setitimer, setpgid, setpgrp, setpgrp, setpriority, setquota, setregid, setreuid, setrlimit, setsid, setsockopt, settimeofday, setuid, shmsys, shutdown, sigblock, sigpause, sigpending, sigsetmask, sigstack, sigsys, sigvec, socket, socketaddr, socketpair, sstk, stat, stat, statfs, stime, stty, swapon, symlink, sync, sysconf, time, times, truncate, umask, umount, uname, unlink, unmount, ustat, utime, utimes, vadvise, vfork, vhangup, vlimit, vpixsys, vread, vtimes, vtrace, vwrite, wait, wait3, wait4, write, writev

(b) Key Windows DLLs and Executables

comctl32
kernel32
msvcpp
msvcrt
mswsock
ntdll
ntoskrnl
user32
ws2_32

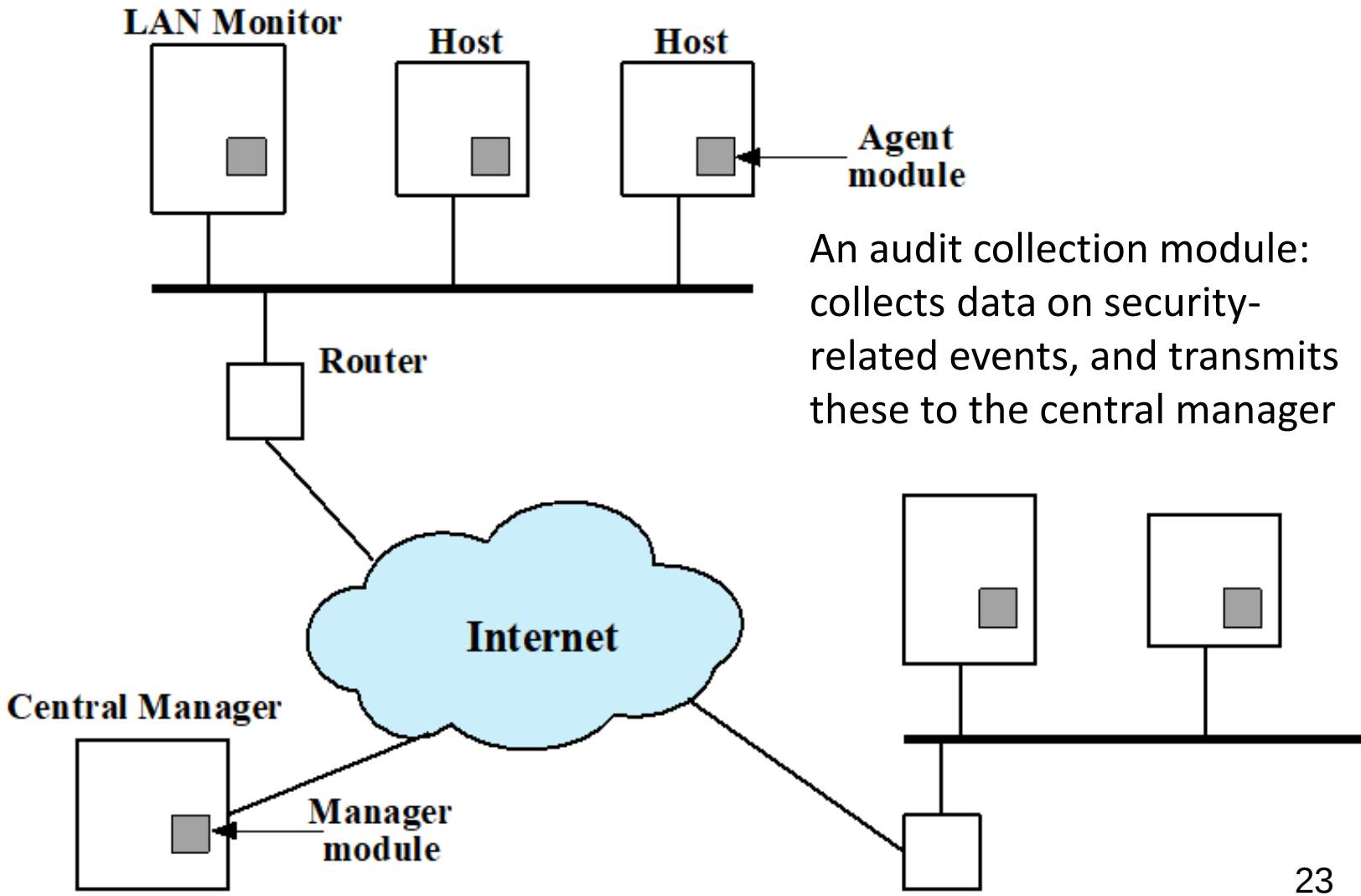
Signature or Heuristic HIDS

- Using a database of
  - File signatures: patterns of data found in known malicious software
  - Heuristic rules: characterizing known malicious behavior
- Widely used in anti-virus software
- Efficient at detecting known malware, but not capable of detecting zero-day attacks

Distributed HIDS

- Major issues
  - Need to deal with different data sensors
  - Assure the integrity and confidentiality of the sensor data
  - Architecture
    - Centralized: bottleneck and single point of failure
    - Decentralized: coordination

Architecture for Distributed Intrusion Detection



An audit collection module: collects data on security-related events, and transmits these to the central manager

Network-based Intrusion Detection (NIDS)

- Monitoring traffic at selected points on a network or interconnected set of networks
  - Packet by packet in real time, or close to real time
  - Network-, transport-, and/or app-level protocol activity

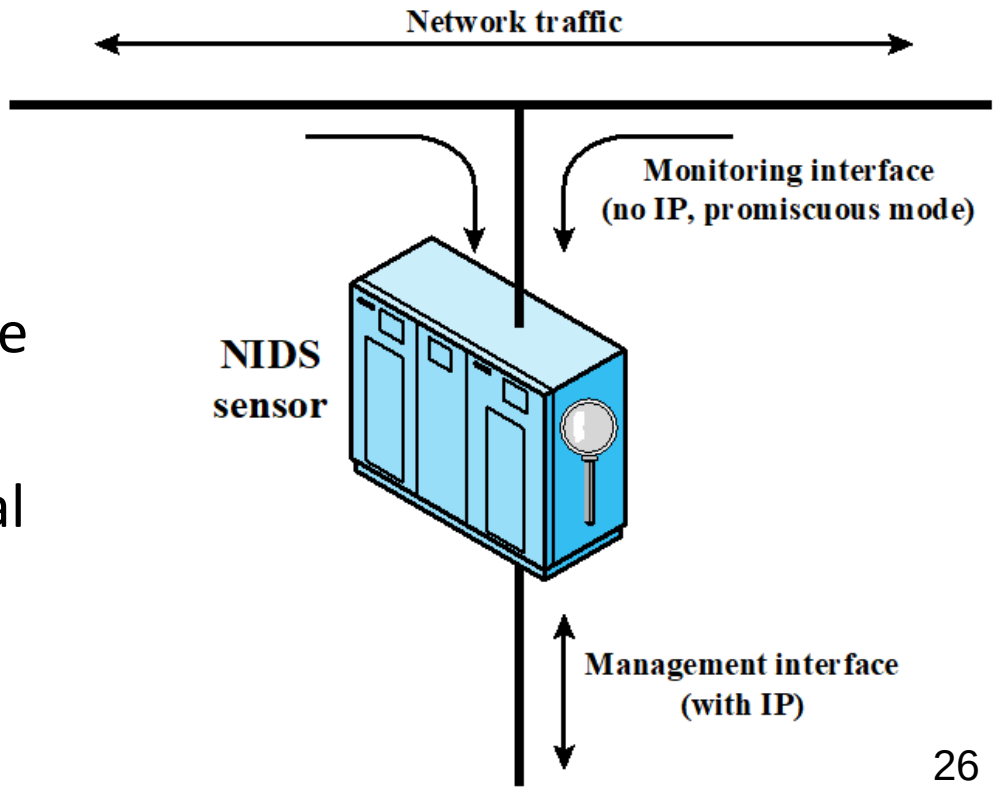
- What is different between NIDS and HIDS?
  - NIDS: examines packet traffic toward potentially vulnerable systems on a network
  - HIDS: examines user and software activity on a host

NIDS (Cont.)

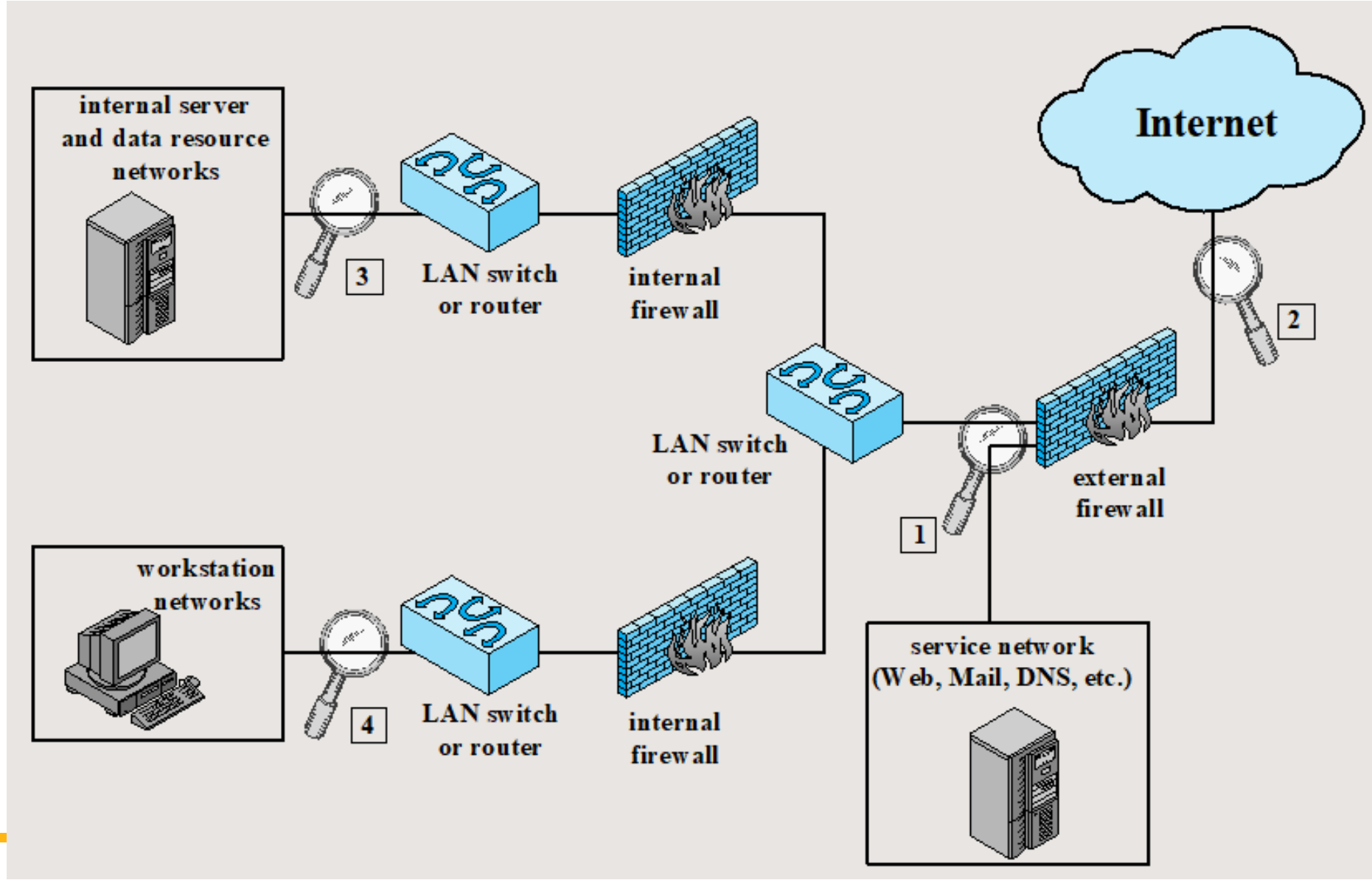
- Typically included in the perimeter security infrastructure of an organization, located with the firewall
- Including
  - A number of sensors to monitor packet traffic
  - One or more servers for management functions
  - One or more management consoles for the human interface
- Their ability gradually becomes to not function well
  - Why? Increasing use of encryption

Two Types of Network Sensors

- Inline sensors
  - Inserted into a network segment
  - Combined with a firewall or a switch
  - Motivation: block an attack when one is detected
  - Pros: no additional separate hardware devices are needed
  - Cons: negative impact on network performance
- Passive sensors
  - Monitoring a copy of network traffic (the actual traffic doesn't pass through)
  - Pros: more efficient and doesn't contribute to packet delay



NIDS Sensor Deployment



Outline

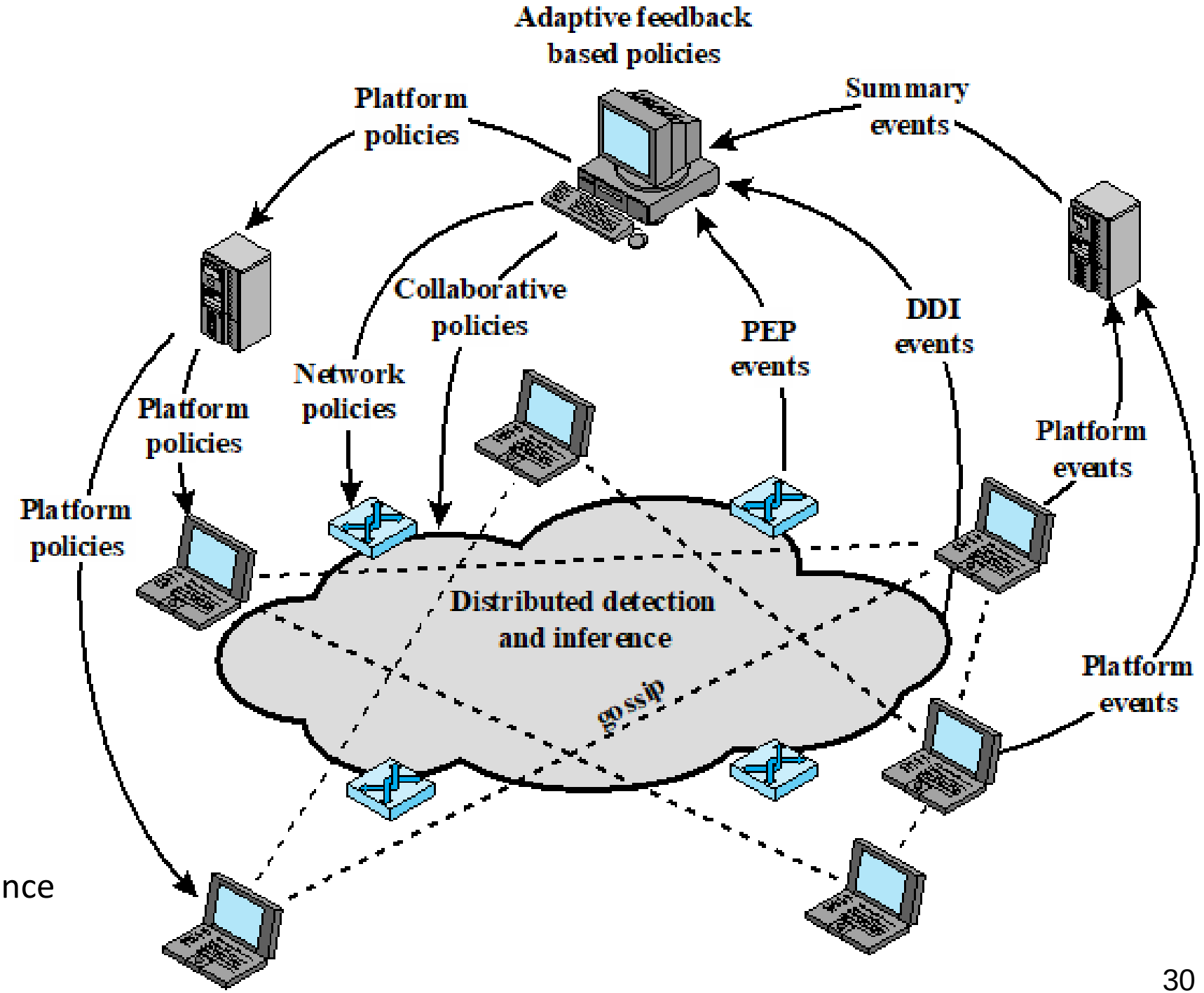
- Intruders
- Intrusion detection
- Analysis approaches
- Host-based intrusion detection
- Network-based intrusion detection
- Distributed or hybrid intrusion detection
- Honeypots

Distributed or Hybrid Intrusion Detection

- Distributed systems that cooperate to identify intrusions and to adapt to changing attack profiles
  - Can recognize attacks based on more subtle clues and then adapt quickly
  - Anomaly detectors at local nodes look for unusual activity

- Due to two key problems confronting IDSs
  - these tools may not recognize new threats or modifications of existing threats
  - it is difficult to update schemes rapidly enough to deal with threats

Overall Architecture of an Automatic Enterprise Security System (by Intel)



PEP: Policy Enforcement Point
DDI: Distributed Detection and Inference

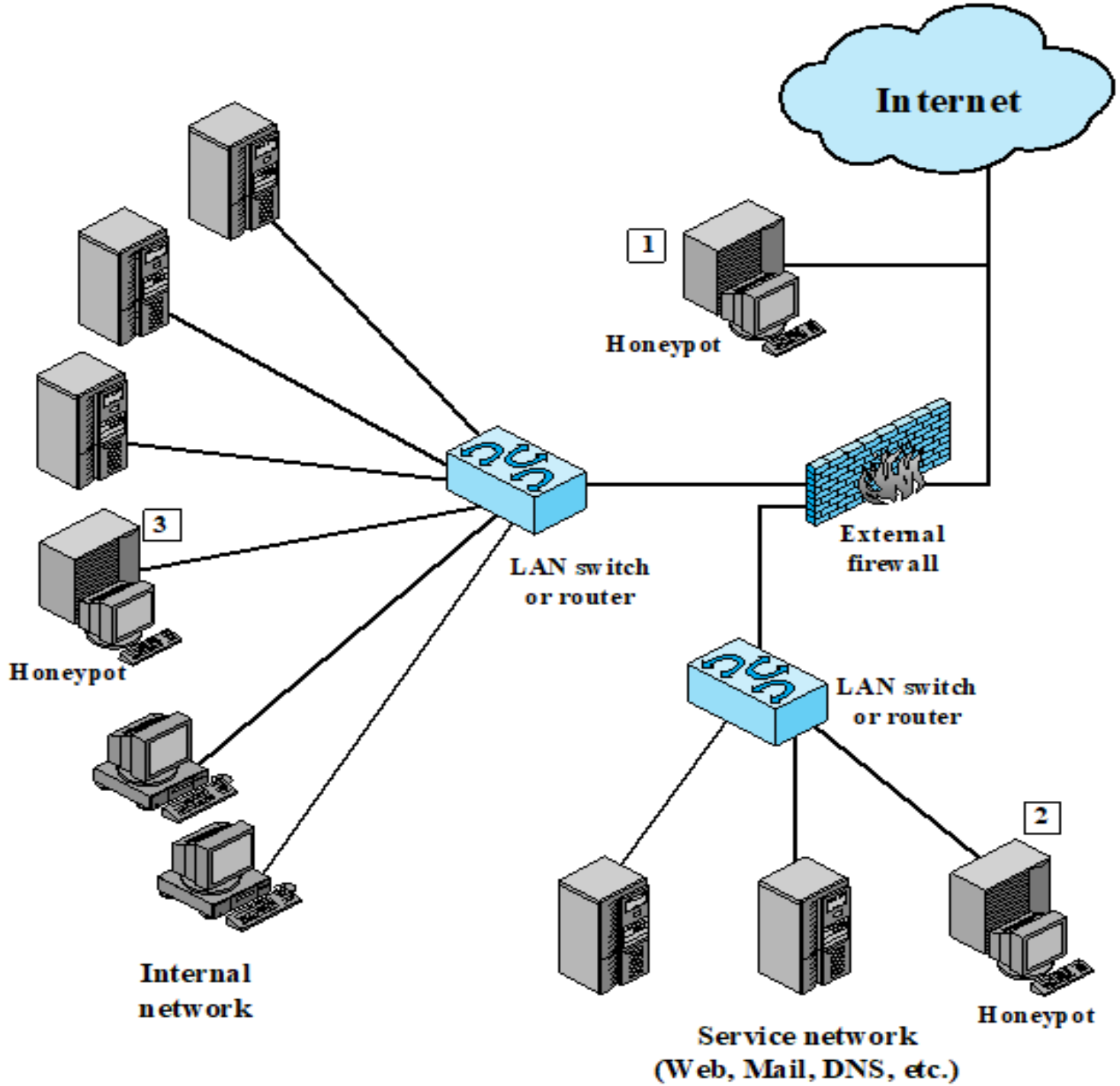
Honeypots

- Decoy systems designed to
  - Lure a potential attacker away from critical systems
  - Collect information about the attacker's activity
  - Encourage the attacker to stay on the system long enough for administrators to respond
- Systems are filled with fabricated information that a legitimate user of the system wouldn't access
- Resources that have no production value
  - Incoming communication is most likely a probe, scan, or attack
  - Initiated outbound communication suggests that the system has probably been compromised

Honeypot Classifications

- Low interaction honeypot
  - Emulating particular IT services or systems well enough to provide a realistic initial interaction, but does not execute a full version
  - Providing a less realistic target
  - Often sufficient for use as a component of a distributed IDS to warn of imminent attack
- High interaction honeypot
  - A real system, with a full OS, services and applications, which are instrumented and deployed where they can be accessed by attackers
  - Is a more realistic target that may occupy an attacker for an extended period
  - However, it requires significantly more resources
  - If compromised, could be used to initiate attacks on other systems

Example of Honeypot Deployment



Questions?