Computer Security: Principles and Practice

Fourth Edition

By: William Stallings and Lawrie Brown

Chapter 8

Intrusion Detection

Classes of Intruders – Cyber Criminals

- **Individuals** or **members of an organized crime group** with a goal of **financial reward**
- Their activities may include:
  - **Identity** theft
  - Theft of **financial credentials**
  - Corporate **espionage**
  - **Data** theft
  - Data **ransoming**
- Typically they are **young**, often Eastern European, Russian, or southeast Asian hackers, who do business on the Web
- They meet in **underground forums** to trade tips and data and coordinate attacks

Classes of Intruders – Activists

- Are either individuals, usually working as **insiders**, or members of a larger group of **outsider attackers**, who are motivated by **social or political causes**
- Also know as **hacktivists**
  - Skill level is often quite low
- Aim of their attacks is often to **promote and publicize their cause** typically through:
  - **Website defacement**
  - **Denial of service** attacks
  - **Theft and distribution of data** that results in negative publicity or compromise of their targets

Classes of Intruders – State-Sponsored Organizations

Groups of hackers sponsored by governments to conduct espionage or sabotage activities

Also known as Advanced Persistent Threats (APTs) due to the covert nature and persistence over extended periods involved with any attacks in this class

Widespread nature and scope of these activities by a wide range of countries from China to the USA, UK, and their intelligence allies

Classes of Intruders – Others

- Hackers with **motivations other than** those previously listed
- Include **classic hackers** or **crackers** who are motivated by **technical challenge** or by **peer-group esteem and reputation**
- Many of those responsible for discovering new categories of **buffer overflow vulnerabilities** could be regarded as members of this class
- Given the wide availability of **attack toolkits**, there is a pool of "**hobby hackers**" using them to explore system and network security

Intruder Skill Levels – Apprentice

- Hackers with **minimal technical skill** who primarily use existing **attack toolkits**
- They likely comprise the largest number of attackers, including many **criminal** and **activist attackers**
- Given their use of existing known tools, these attackers are the **easiest to defend against**
- Also known as "**script-kiddies**" due to their use of **existing scripts (tools)**

Intruder Skill Levels – Journeyman

- Hackers with **sufficient technical skills** to **modify and extend attack toolkits** to use newly discovered, or purchased, vulnerabilities
- They may be able to **locate new vulnerabilities** to exploit that are **similar to some already known**
- Hackers with such skills are likely found **in all intruder classes**
- **Adapt tools** for use by others

Intruder Skill Levels – Master

- Hackers with **high-level technical skills** capable of discovering **brand new categories of vulnerabilities**
- **Write new powerful attack toolkits**
- Some of the **better known classical hackers** are of this level
- Some are employed by **state-sponsored organizations**
- Defending against these attacks is **of the highest difficulty**

Examples of Intrusion

- Remote **root** compromise
- **Web** server defacement
- Guessing/cracking **passwords**
- Copying databases containing **credit card numbers**
- Viewing **sensitive data** without authorization
- Running a **packet** sniffer
- Distributing **pirated software**
- Using an unsecured modem to access **internal network**
- Impersonating an executive to get **information**
- Using an **unattended workstation**

Intruder Behavior

Target acquisition and information gathering	**Initial access**	**Privilege escalation**
Information gathering or system exploit	**Maintaining access**	**Covering tracks**

(a) Target Acquisition and Information Gathering

- Explore corporate website for information on corporate structure, personnel, key systems, as well as details of specific web server and OS used.
- Gather information on target network using DNS lookup tools such as dig, host, and others; and query WHOIS database.
- Map network for accessible services using tools such as NMAP.
- Send query email to customer service contact, review response for information on mail client, server, and OS used, and also details of person responding.
- Identify potentially vulnerable services, eg vulnerable web CMS.

(b) Initial Access

- Brute force (guess) a user's web content management system (CMS) password.
- Exploit vulnerability in web CMS plugin to gain system access.
- Send spear-phishing email with link to web browser exploit to key people.

(c) Privilege Escalation

- Scan system for applications with local exploit.
- Exploit any vulnerable application to gain elevated privileges.
- Install sniffers to capture administrator passwords.
- Use captured administrator password to access privileged information.

(d) Information Gathering or System Exploit

- Scan files for desired information.
- Transfer large numbers of documents to external repository.
- Use guessed or captured passwords to access other servers on network.

(e) Maintaining Access

- Install remote administration tool or rootkit with backdoor for later access.
- Use administrator password to later access network.
- Modify or disable anti-virus or IDS programs running on system.

(f) Covering Tracks

- Use rootkit to hide files installed on system.
- Edit logfiles to remove entries generated during the intrusion.

Table 8.1

Examples of Intruder Behavior

(Table can be found on pages 255-256 in the textbook.)

Definitions

- 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)
  - Monitors the characteristics of **a single host** for suspicious activity
- Network-based IDS (NIDS)
  - Monitors **network traffic** and analyzes **network**, **transport**, and **application protocols** to identify suspicious activity
- Distributed or hybrid IDS
  - Combines information from a number of sensors, often **both host and network based**, in a central analyzer that is able to better identify and respond to intrusion activity

Comprises three logical components:

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

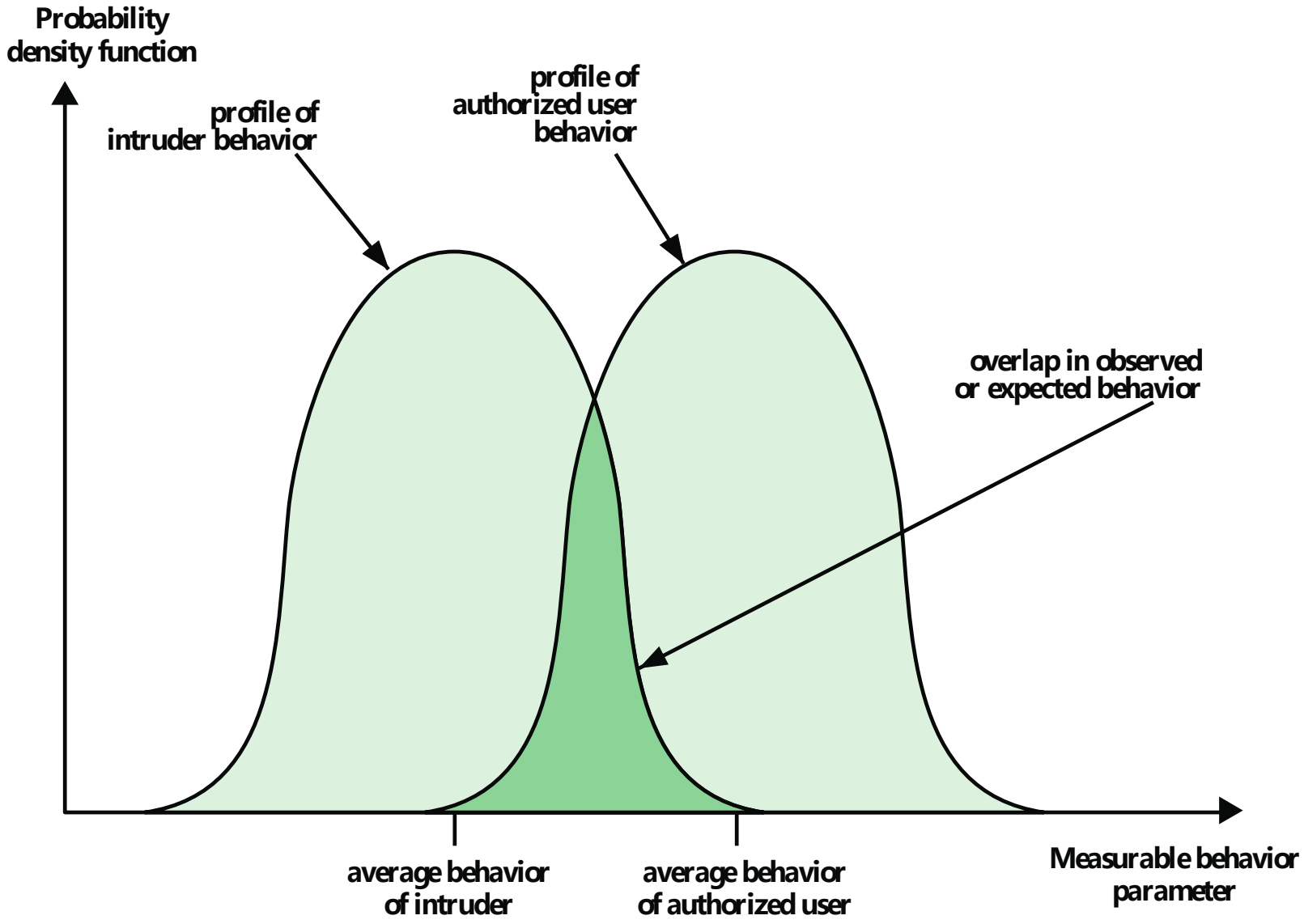

Figure 8.1 Profiles of Behavior of Intruders and Authorized Users

IDS 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

- Involves the **collection of data** relating to the behavior of **legitimate users** over a period of time
- Current observed behavior is analyzed to **determine whether** this behavior is that of a **legitimate user** or that of **an intruder**

Signature/Heuristic detection

- Uses **a set of known malicious data patterns or attack rules** that are compared with current behavior
- Also known as **misuse detection**
- Can only identify **known attacks** for which it has patterns or rules

Anomaly Detection

A variety of classification approaches are used:

Statistical	Knowledge based	Machine-learning
• Analysis of the observed behavior using **univariate**, **multivariate**, or **time-series models** of observed metrics	• Approaches use an **expert system** that classifies observed behavior according to **a set of rules** that model legitimate behavior	• Approaches automatically determine a suitable **classification model** from the training data using data mining techniques

Signature or Heuristic Detection

Signature approaches

Match a large collection of known patterns of malicious data against data stored on a system or in transit over a network

The signatures need to be large enough to minimize the false alarm rate, while still detecting a sufficiently large fraction of malicious data

Widely used in anti-virus products, network traffic scanning proxies, and in NIDS

Rule-based heuristic identification

Involves the use of rules for identifying known penetrations or penetrations that would exploit known weaknesses

Rules can also be defined that identify suspicious behavior, even when the behavior is within the bounds of established patterns of usage

Typically rules used are specific

SNORT is an example of a rule-based NIDS

Host-Based Intrusion Detection (HIDS)

- Adds a **specialized layer of security software** to vulnerable or sensitive systems
- Can use either **anomaly** or **signature and heuristic approaches**
- Monitors activity to **detect suspicious behavior**
  - Primary purpose is to **detect intrusions**, **log suspicious events,** and **send alerts**
  - Can detect both **external** and **internal intrusions**

Data Sources and Sensors

A fundamental component of intrusion detection is the **sensor** that collects data

Common **data sources** include:

- **System call** traces
- Audit (**log file**) records
- **File integrity** checksums
- **Registry** access

Table 8.2

Linux System Calls and Windows DLLs Monitored

(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

(Table can be found on page 264 in the textbook)

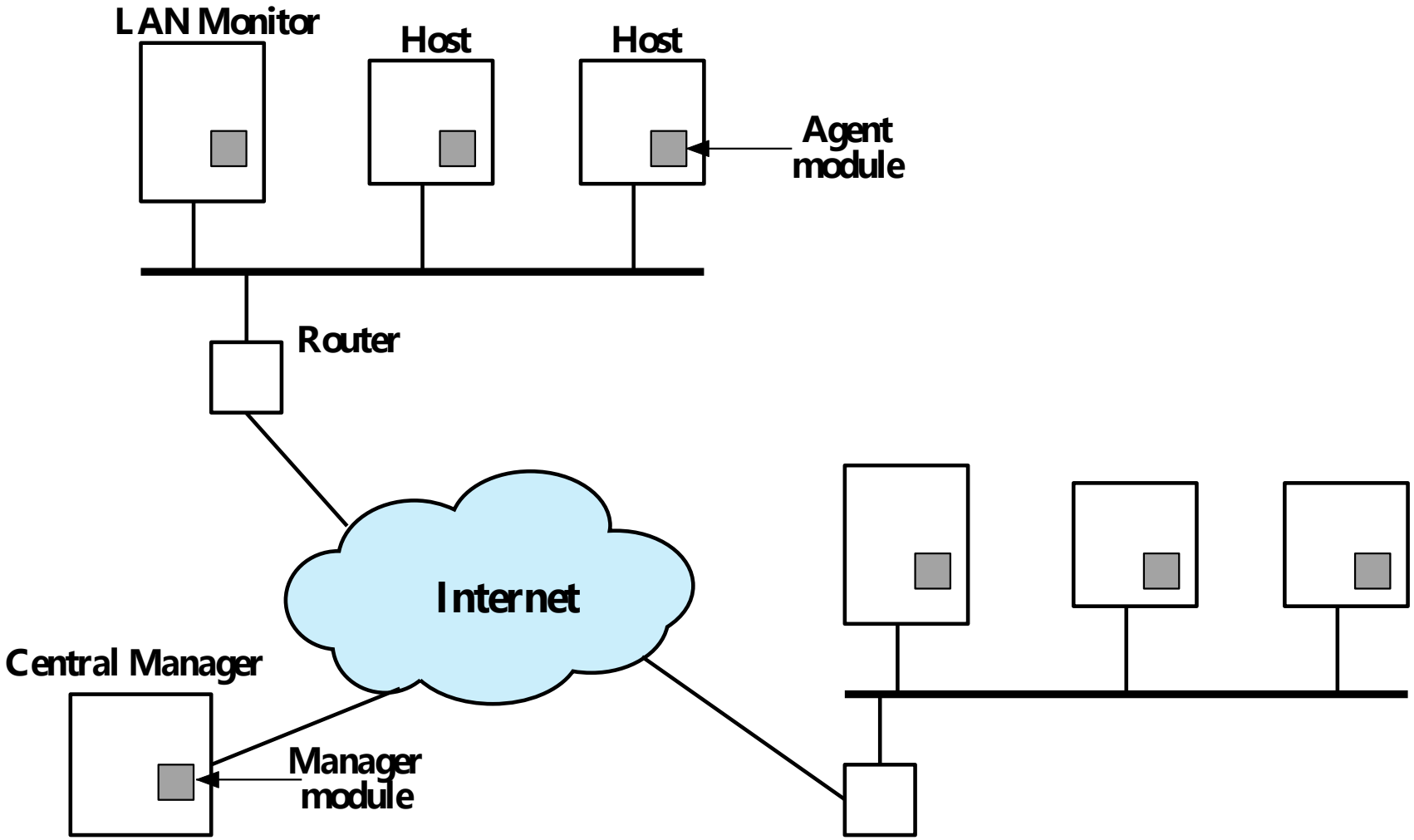

Figure 8.2 Architecture for Distributed Intrusion Detection

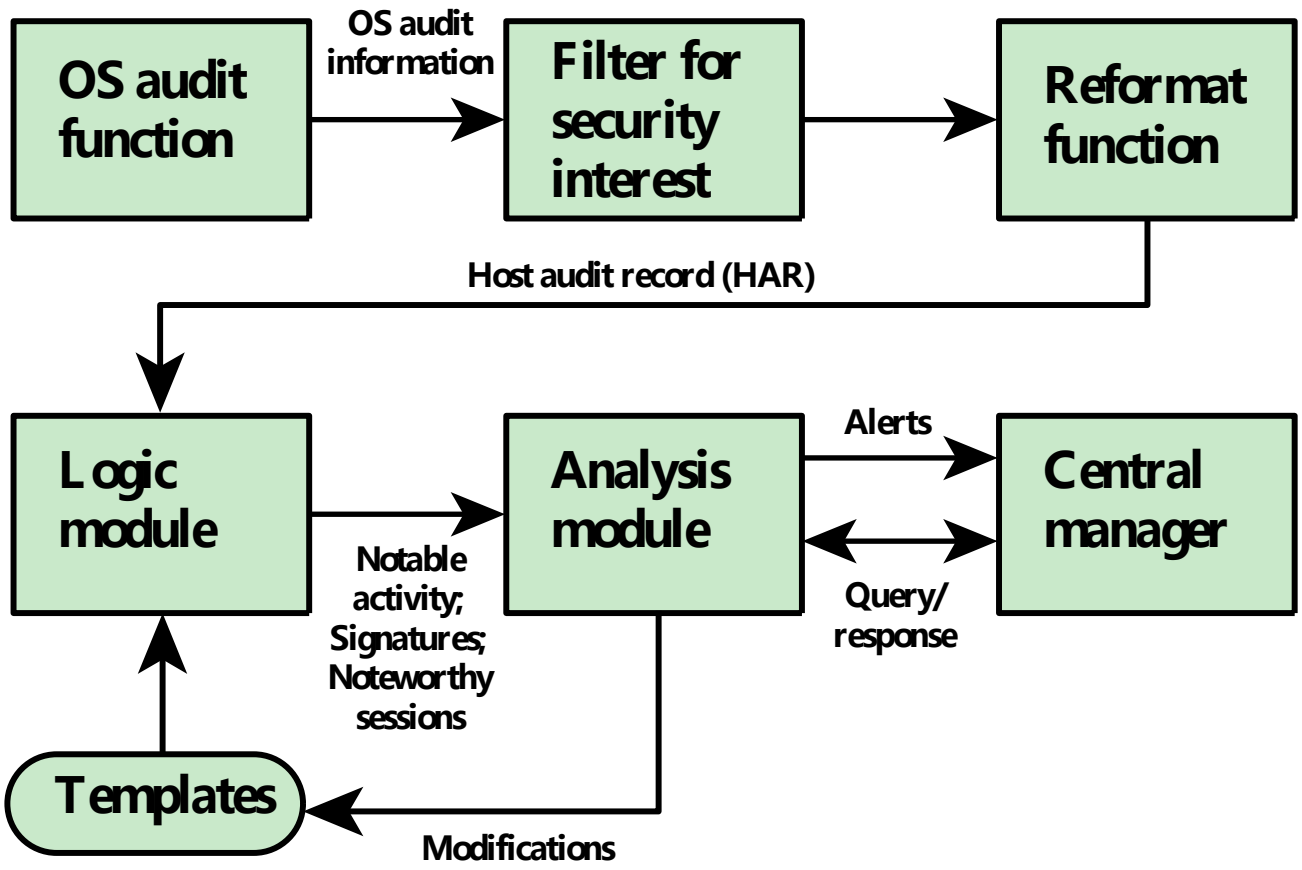

Figure 8.3 Agent Architecture

Network-Based IDS (NIDS)

Monitors traffic at selected points on a network

Examines traffic packet by packet in real or close to real time

May **examine** network, transport, and/or application-level **protocol activity**

Comprised of a number of **sensors**, one or more servers for **NIDS management** functions, and one or more management consoles for the **human interface**

Analysis of traffic patterns may be done at the **sensor**, the **management server** or a **combination** of the two

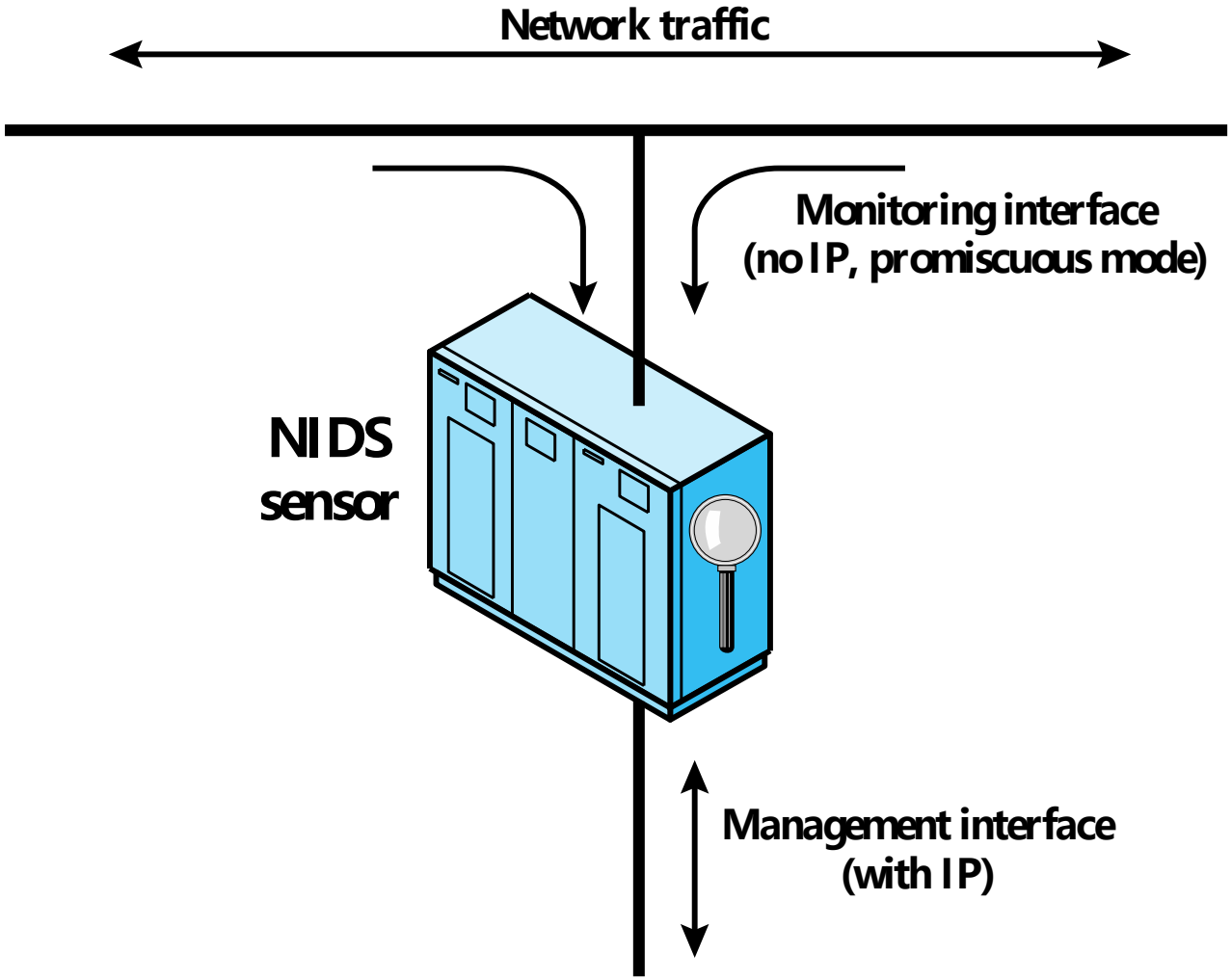

Figure 8.4 Passive NIDS Sensor

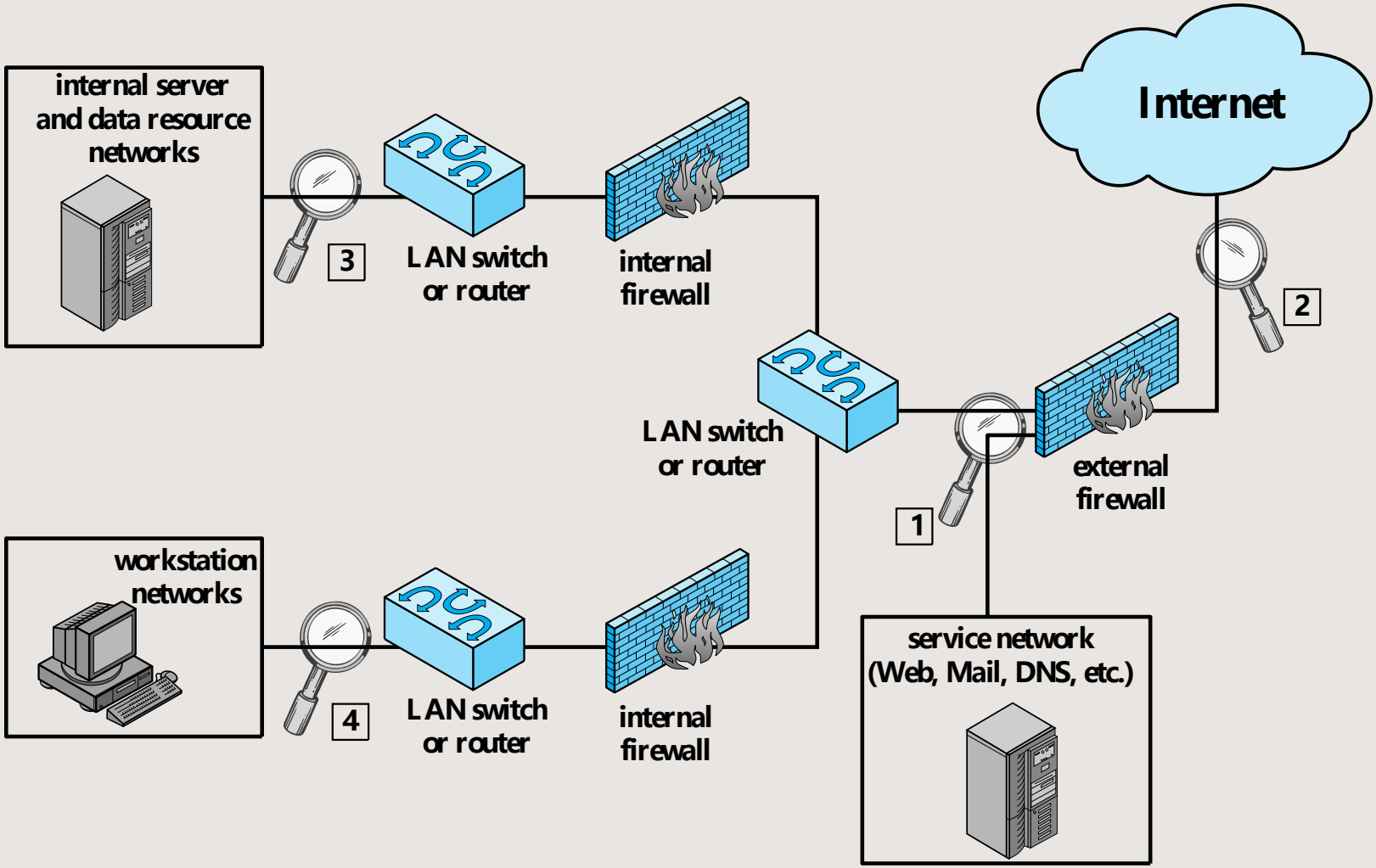

Figure 8.5 Example of NIDS Sensor Deployment

Intrusion Detection Techniques

Attacks suitable for Signature detection

- **Application layer** reconnaissance and attacks
- **Transport layer** reconnaissance and attacks
- **Network layer** reconnaissance and attacks
- **Unexpected application services**
- **Policy violations**

Attacks suitable for Anomaly detection

- Denial-of-service (**DoS**) attacks
- **Scanning**
- **Worms**

Stateful Protocol Analysis (SPA)

- **Subset of anomaly detection** that compares observed network traffic against **predetermined universal vendor supplied profiles** of benign protocol traffic
  - This distinguishes it from anomaly techniques trained with organization specific traffic protocols
- **Understands** and **tracks** network, transport, and application protocol **states** to ensure they progress as expected
- A key disadvantage is the **high resource use** it requires

Logging of Alerts

- **Typical information** logged by a NIDS sensor includes:
  - **Timestamp**
  - **Connection** or **session** ID
  - **Event** or **alert** type
  - **Rating**
  - Network, transport, and application layer **protocols**
  - Source and destination **IP addresses**
  - Source and destination **TCP** or **UDP** ports, or **ICMP** types and codes
  - **Number of bytes** transmitted over the connection
  - **Decoded payload data**, such as application requests and responses
  - **State-related information**

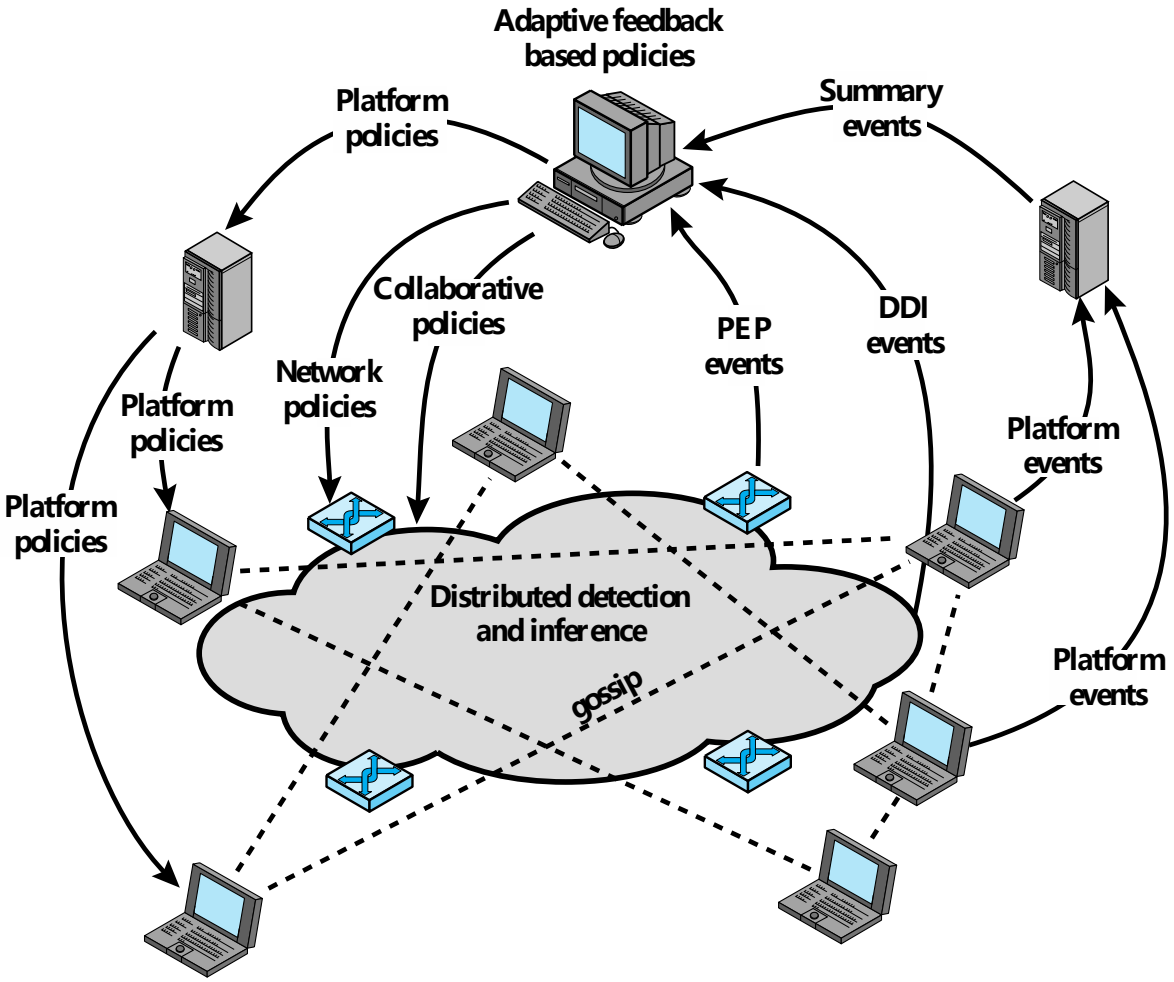

Figure 8.6 Overall Architecture of an Autonomic Enterprise Security System

IETF Intrusion Detection Working Group

- Purpose is to define **data formats** and **exchange procedures** for sharing information of interest to **intrusion detection** and **response systems** and to **management systems** that may need to interact with them
- The working group issued the following RFCs in 2007:

Intrusion Detection Message Exchange Requirements (RFC 4766)

- Document defines requirements for the Intrusion Detection Message Exchange Format (IDMEF)
- Also specifies requirements for a communication protocol for communicating IDMEF

The Intrusion Detection Message Exchange Format (RFC 4765)

- Document describes a data model to represent information exported by intrusion detection systems and explains the rationale for using this model
- An implementation of the data model in the Extensible Markup Language (XML) is presented, and XML Document Type Definition is developed, and examples are provided

The Intrusion Detection Exchange Protocol (RFC 4767)

- Document describes the Intrusion Detection Exchange Protocol (IDXP), an application level protocol for exchanging data between intrusion detection entities
- IDXP supports mutual authentication, integrity, and confidentiality over a connection oriented protocol

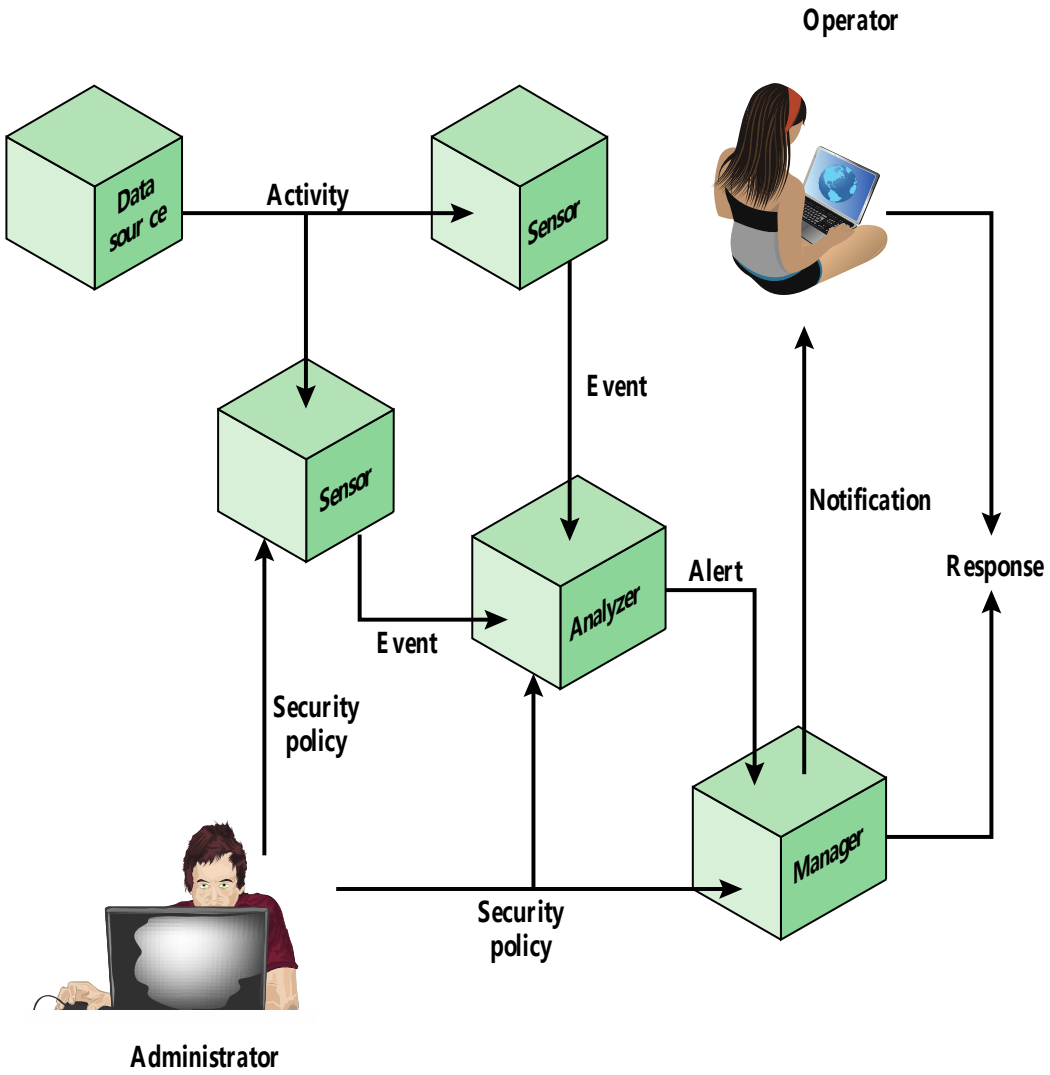

Figure 8.7 Model For Intrusion Detection Message Exchange

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**
  - Therefore **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
  - Consists of a software package that emulates particular IT services or systems well enough to provide a **realistic initial interaction**, but does not execute a full version of those services or systems
  - Provides 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 operating system, 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

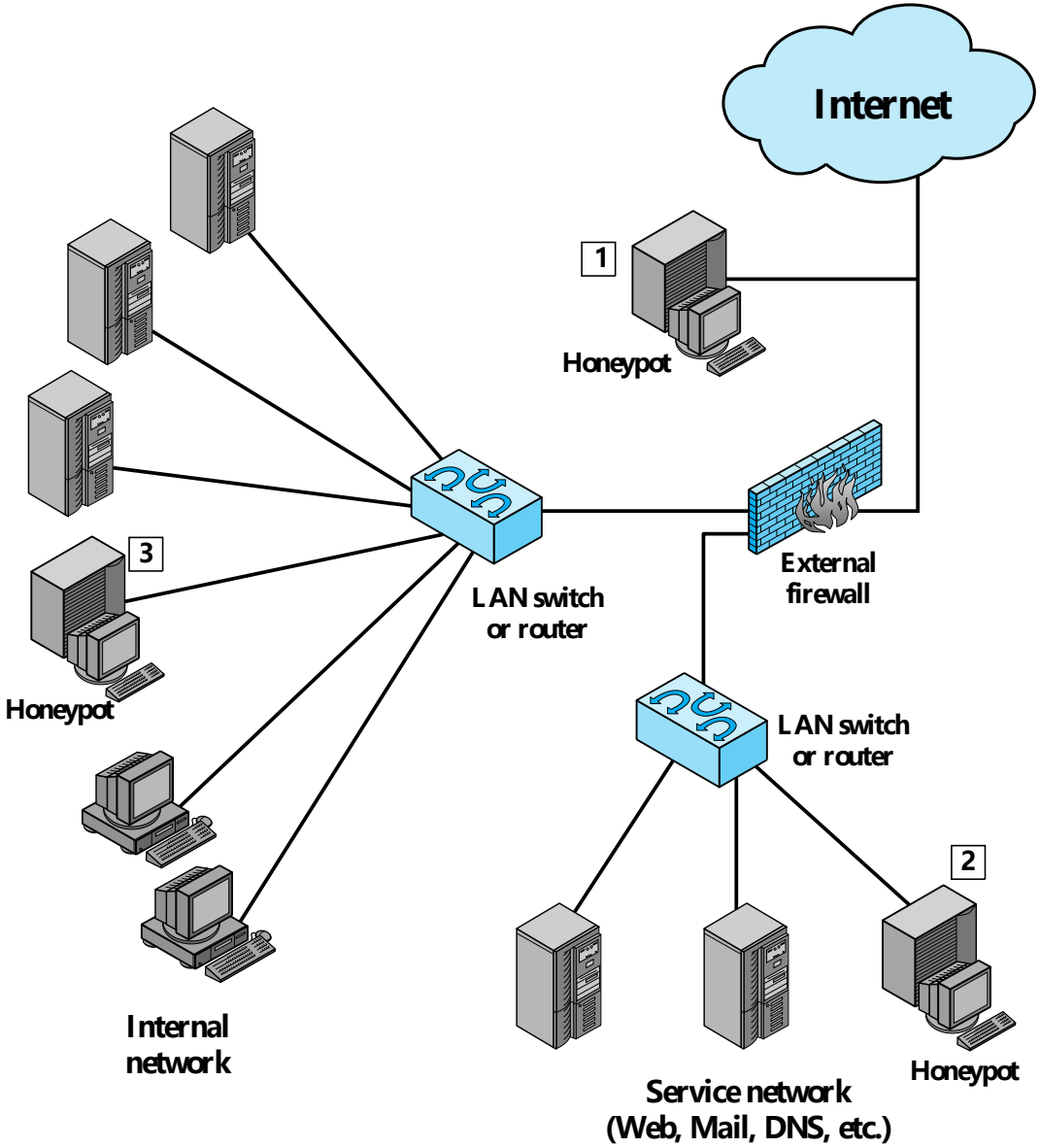

Figure 8.8 Example of Honeypot Deployment

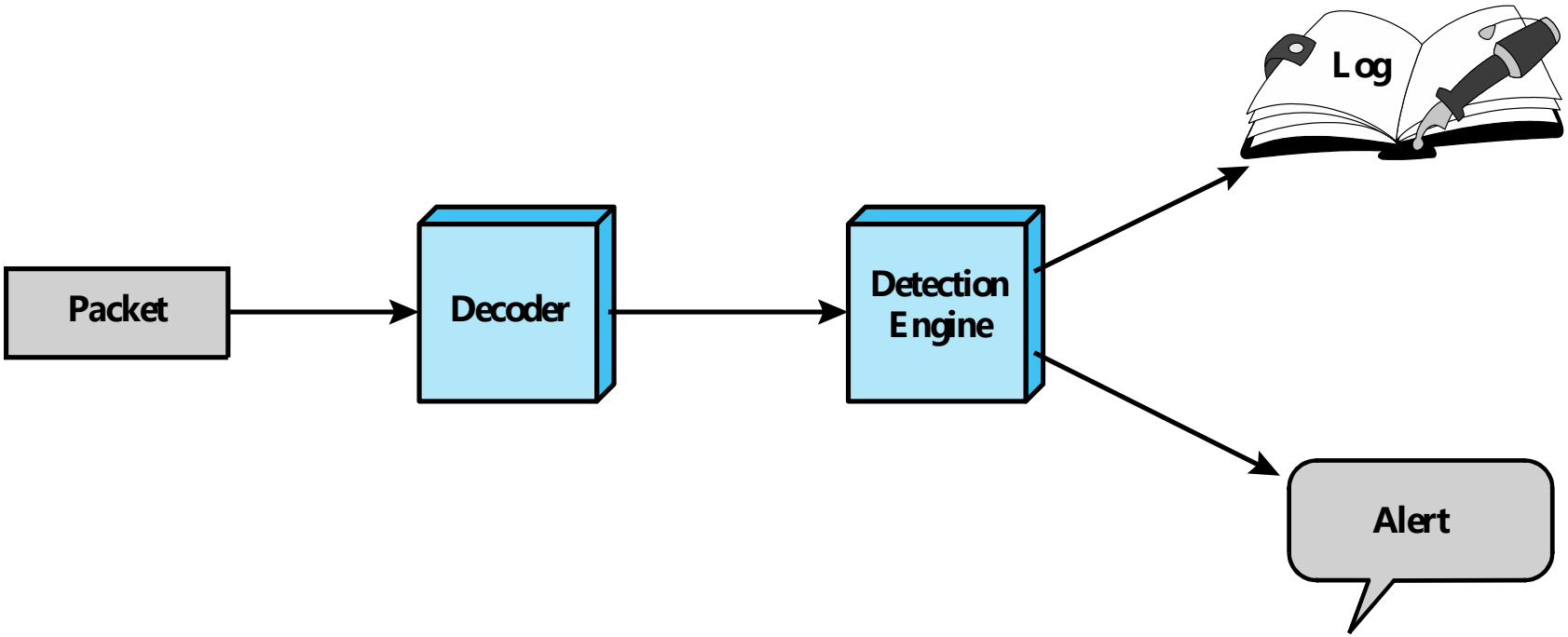

Figure 8.9 Snort Architecture

Summary

- Intruders
  - Intruder behavior
- Intrusion detection
  - Basic principles
  - The base-rate fallacy
  - Requirements
- Analysis approaches
  - Anomaly detection
  - Signature or heuristic detection
- Distributed or hybrid intrusion detection
- Intrusion detection exchange format
- Honeypots
- Host-based intrusion detection
  - Data sources and sensors
  - Anomaly HIDS
  - Signature or heuristic HIDS
  - Distributed HIDS
- Network-based intrusion detection
  - Types of network sensors
  - NIDS sensor deployment
  - Intrusion detection techniques
  - Logging of alerts
- Example system: Snort
  - Snort architecture
  - Snort rules

作业

- 英文教材（第四版）P306-308
- Questions 8.10, 8.14
- Problems 8.3, 8.7