Chapter 8 (8.1 to 8.6)

Classes of Intruders:

- **Cyber Criminals -** Individuals or members of an organized crime group with a goal of financial reward
- **Activists -** individuals, usually working as insiders, or members of a larger group of outsider attackers, who are motivated by social or political causes. Also known as hacktivists. Aim of their attacks is often to promote and publicize their cause typically through: Website defacement, DDoS attack.
- **State-Sponsored Organizations -** Groups of hackers sponsored by governments to conduct espionage or sabotage activities
- **Others –** Hackers with motivation other than mentioned above. Include classic hackers or crackers who are motivated by technical challenge or by peer-group esteem and reputation. Those who discover new categories of buffer overflow vulnerabilities.

Intruder Skill Levels:

- **Apprentice -** Hackers with minimal technical skill who primarily use existing attack toolkits. These attackers are easiest to defend against. Also known as "script-kiddies" due to their use of existing scripts (tools)
- **Journeyman -** Hackers with sufficient technical skills to modify and extend attack toolkits to use newly discovered, or purchased, vulnerabilities
- **Master -** Hackers with high-level technical skills capable of discovering brand new categories of vulnerabilities. Write new powerful attack toolkits. Some are employed by state-sponsored organizations. Defending against these attacks is of the highest difficulty.

Examples of Intrusions:

- 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:

1. **Target Acquisition and Information Gathering:** Where the attacker identifies and characterizes the target systems using publicly available information, both technical and non-technical, and then use network exploration tools to map target resources.
2. **Initial Access:** The initial access to a target system, typically by exploiting a remote network vulnerability by guessing weak authentication credentials used in a remote service, or via the

installation of malware on the system using some form of social engineering or drive-by-download attack.

3. **Privilege Escalation:** Actions taken on the system, typically via a local access vulnerability to increase the privileges available to the attacker to enable their desired goals on the target system.
4. **System Exploit:** Actions by the attacker to access or modify information or resources on the system, or to navigate to another target system.
5. **Maintaining Access:** Actions such as the installation of backdoors or other malicious software, or through the addition of covert authentication credentials or other configuration changes to the system, to enable continued access by the attacker after the initial attack.
6. **Covering Tracks:** Where the attacker disables or edits audit logs, to remove evidence of attack activity, and uses rootkits and other measures to hide covertly installed files or code.

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:

An IDS comprises three logical components:

1. **Sensors**: Sensors are responsible for collecting data. The input for a sensor may be any part of a system that could contain evidence of an intrusion. Types of input to a sensor includes network packets, log files, and system call traces. Sensors collect and forward this information to the analyzer.
2. **Analyzers**: Analyzers receive input from one or more sensors or from other analyzers. The analyzer is responsible for determining if an intrusion has occurred.
3. **User interface**: The user interface to an IDS enables a user to view output from the system or control the behavior of the system.

IDSs are often classified based on the source and type of data analyzed, as:

1. **Host-based IDS (HIDS):** Monitors the characteristics of a single host and the events occurring within that host, such as process identifiers and the system calls they make, for evidence of suspicious activity. Can detect both external and internal intrusions
2. **Network-based IDS (NIDS):** Monitors network traffic for particular network segments or devices and analyzes network, transport, and application protocols to identify suspicious activity.
3. **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.

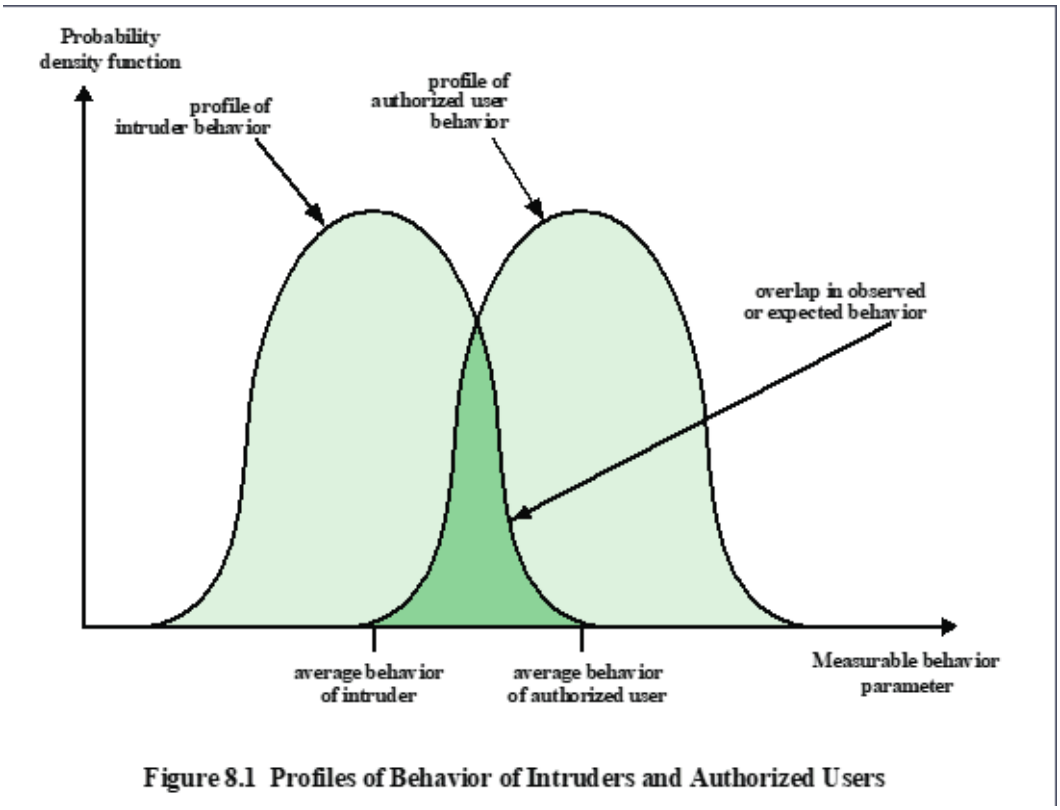


Figure 8.1 Profiles of Behavior of Intruders and Authorized Users

Although the typical behavior of an intruder differs from the typical behavior of an authorized user, there is an overlap in these behaviors. Thus, a loose interpretation of intruder behavior, which will catch more intruders, will also lead to a number of **false positives**, or false alarms, where authorized users are identified as intruders. On the other hand, an attempt to limit false positives by a tight interpretation of intruder behavior will lead to an increase in **false negatives**, or intruders not identified as intruders. Thus, there is an element of compromise and art in the practice of intrusion detection. Ideally you want an IDS to have a high detection rate, that is, the ratio of detected to total attacks, while minimizing the false alarm rate, the ratio of incorrectly classified to total normal usage.

An intrusion detection system has a high false positive rate, it may generate a large number of false alarms, which can be costly and time-consuming to investigate. Additionally, if the system has a low true positive rate, it may fail to detect real intrusions, which can have serious security implications. This is an example of a phenomenon known as the **base-rate fallacy**

IDS Requirements:

1. **Run continually** with minimal human supervision.
2. Be **fault tolerant** in the sense that it must be able to recover from system crashes and initializations.
3. **Resist subversion**. The IDS must be able to monitor itself and detect if it has been modified by an attacker.
4. Impose a **minimal overhead on the system** where it is running.
5. Be able to be **configured according to the security policies** of the system that is being monitored.
6. Be able **to adapt to changes in system and user behavior** over time.
7. Be able to **scale to monitor a large number of hosts**.
8. **Provide graceful degradation of service** in the sense that if some components of the IDS stop working for any reason, the rest of them should be affected as little as possible.
9. **Allow dynamic reconfiguration**; that is, the ability to reconfigure the IDS without having to restart it.

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

Anomaly Detection can be categorized as:

1) Statistical approaches use the captured sensor data to develop a statistical profile of the observed metrics. Univariate models, where each metric was treated as an independent random variable. Multivariate models considered correlations between the metrics, Time-series models use the order and time between observed events to better classify the behavior.

The **advantages** of these statistical approaches include their relative simplicity and low computation cost, and lack of assumptions about behavior expected.

Their **disadvantages** include the difficulty in selecting suitable metrics to obtain a reasonable balance between false positives and false negatives, and that not all behaviors can be modeled using these approaches.

2) Knowledge based approaches classify the observed data using a set of rules. These rules are developed during the training phase, usually manually, to characterize the observed training data into distinct classes.

The **advantages** of knowledge-based approaches include their robustness and flexibility.

Their main **disadvantage** is the difficulty and time required to develop high-quality knowledge from the data, and the need for human experts to assist with this process.

3) Machine-learning approaches use data mining techniques to automatically develop a model using the labeled normal training data. This model is then able to classify subsequently observed data as either normal or anomalous

The **advantages** of the machine-learning approaches include their flexibility, adaptability, and ability to capture interdependencies between the observed metrics.

A key **disadvantage** is that this process typically requires significant time and computational resources.

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

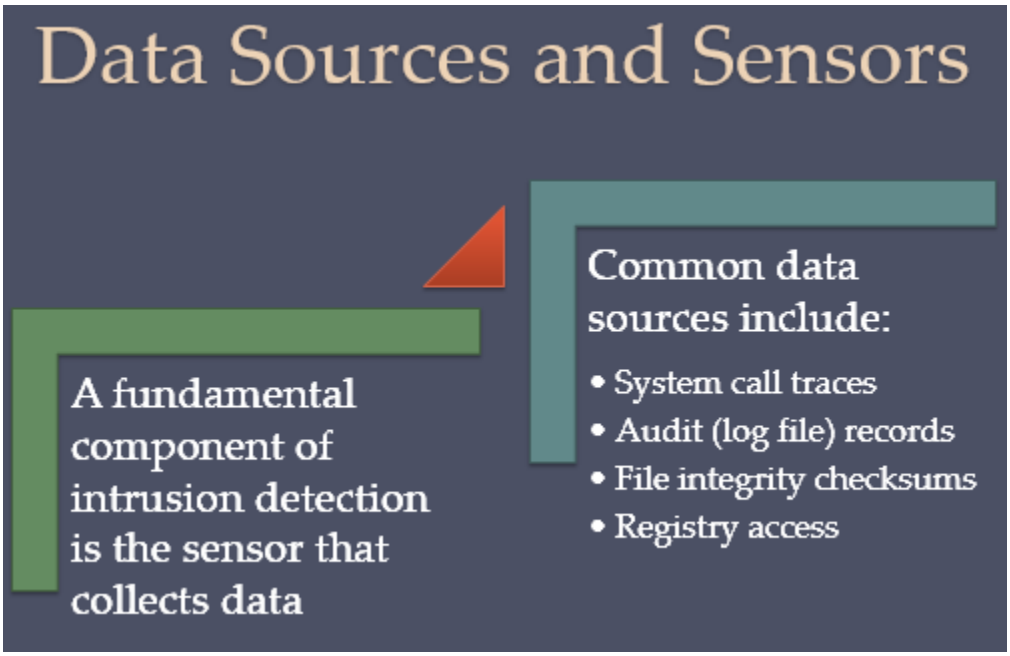
Signature/Heuristic detection can be categorized as:

1) Signature-based intrusion detection techniques use a database of known attack patterns to identify potential intrusions.

Some **advantages** of signature-based techniques include their accuracy and ability to quickly detect known attacks, while some **disadvantages** include their inability to detect previously unknown attacks and the need for a constantly updated database of attack signatures.

2) Heuristic-based intrusion detection techniques use algorithms to identify potential intrusions based on certain characteristics or behaviors that are indicative of an attack.

Some **advantages** of heuristic-based techniques include their flexibility and ability to detect previously unknown attacks, while some **disadvantages** include their lower accuracy and higher rate of false positives.



System calls are the means by which programs access core kernel functions, providing a wide range of interactions with the low-level operating system functions. Hence they provide detailed information on process activity that can be used to classify it as normal or anomalous

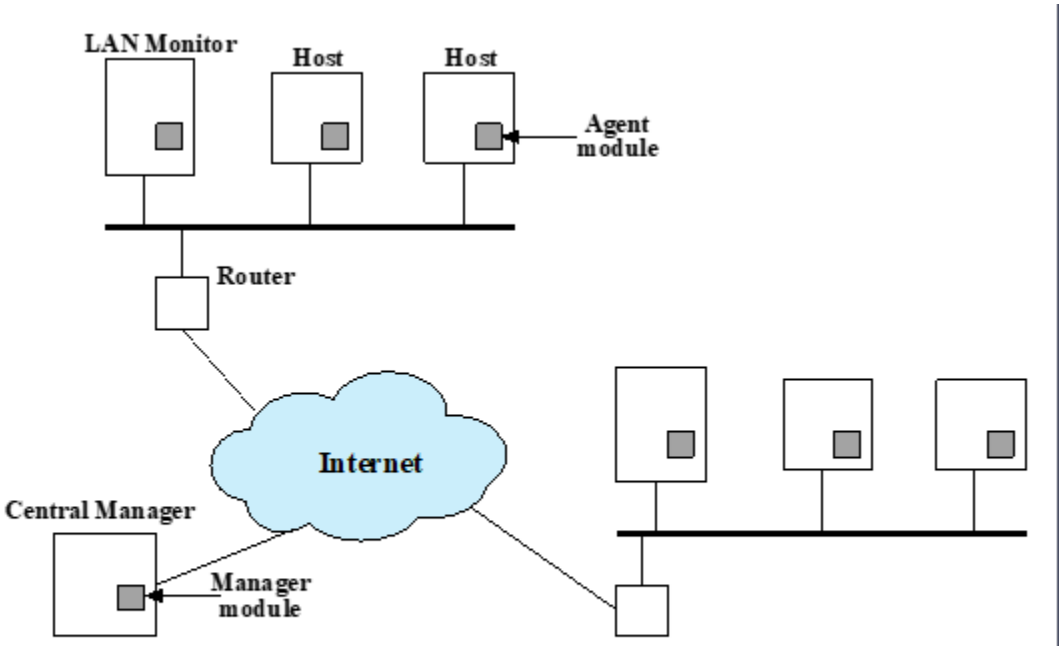


Figure 8.2 Architecture for Distributed Intrusion Detection

1. **Host agent module**: An audit collection module operating as a background process on a monitored system. Its purpose is to collect data on security-related events on the host and transmit these to the central manager.

2. **LAN monitor agent module:** Operates in the same fashion as a host agent module except that it analyzes LAN traffic and reports the results to the central manager.

3. **Central manager module**: Receives reports from LAN monitor and host agents and processes and correlates these reports to detect intrusion.

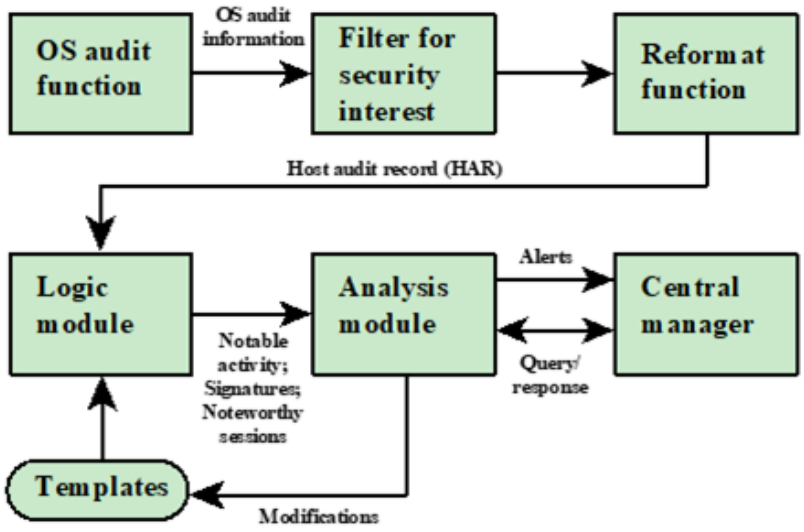
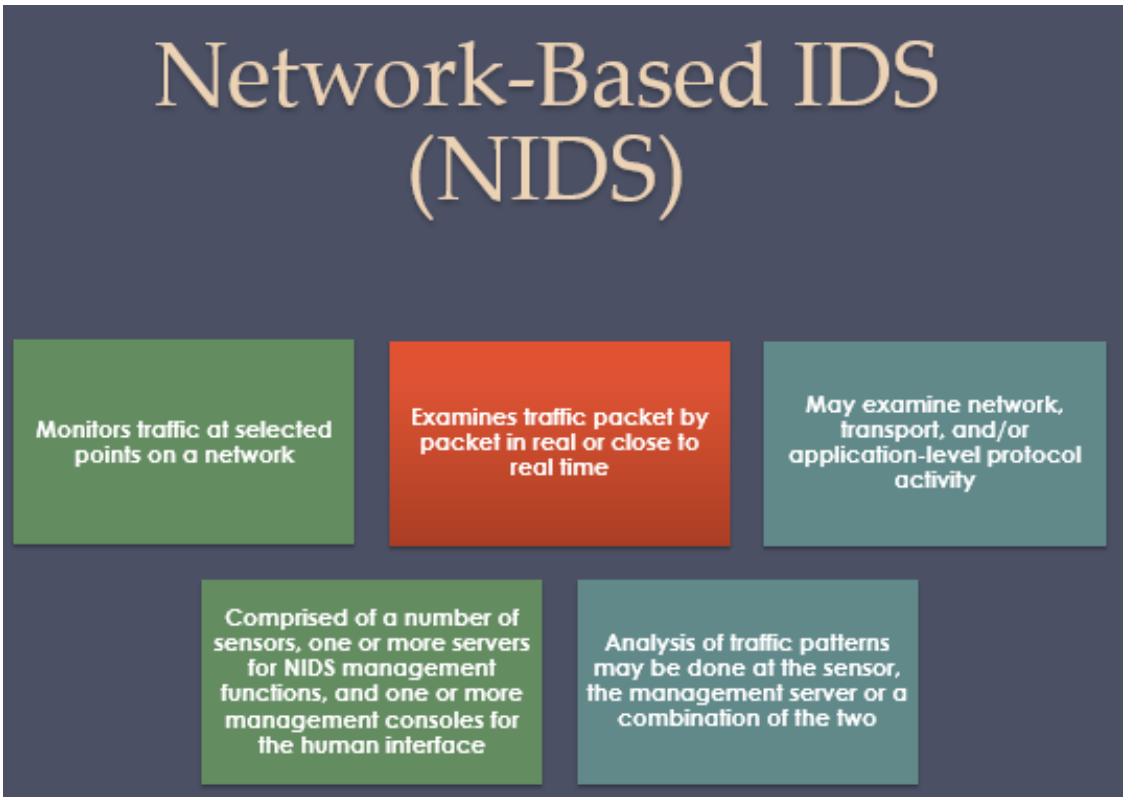


Figure 8.3 Agent Architecture

Figure 8.3 shows the general approach that is taken. The agent **captures each audit record** produced by the native audit collection system. A **filter** is applied that retains only those records that are of **security interest**. These records are then **reformatted into a standardized format** referred to as the **host audit record (HAR)**. Next, a template-driven **logic module** analyzes the records for suspicious activity. At the lowest level, the **agent scans for notable events that are of interest independent of any past events**. At the next higher level, the **agent looks for sequences of events**, such as known attack patterns (signatures). Finally, the agent looks for **anomalous behavior** of an individual user based on a historical profile of that user, such as number of programs executed, number of files accessed, and the like. When suspicious activity is detected, an alert is sent to the central manager. The central manager includes an expert system that can draw inferences from received data. The manager may also query individual systems for copies of HARs to correlate with those from other agents.



An **inline sensor** is inserted into a network segment so that the traffic that it is monitoring must pass through the sensor. This approach has the advantage that no additional separate hardware devices are needed; all that is required is NIDS sensor software. The device is performing both intrusion detection and intrusion prevention functions.

More commonly, **passive sensors** are used. A passive sensor monitors a copy of network traffic; the actual traffic does not pass through the device. From the point of view of traffic flow, the passive sensor is more efficient than the inline sensor, because it does not add an extra handling step that contributes to packet delay.

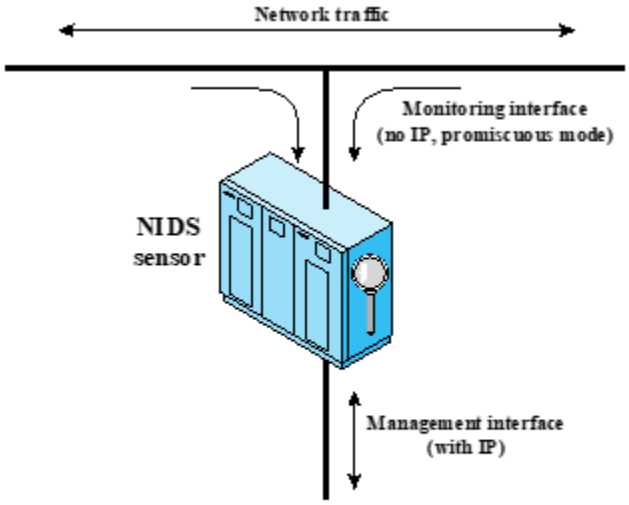


Figure 8.4 illustrates a typical passive sensor configuration. The sensor connects to the network transmission medium, such as a fiber optic cable, by a direct physical tap. The tap provides the sensor with a copy of all network traffic being carried by the medium. The network interface card (NIC) for this tap usually does not have an IP address configured for it. All traffic into this NIC is simply collected with no protocol interaction with the network. The sensor has a second NIC that connects to the network with an IP address and enables the sensor to communicate with a NIDS management server.

Figure 8.4 Passive NIDS Sensor

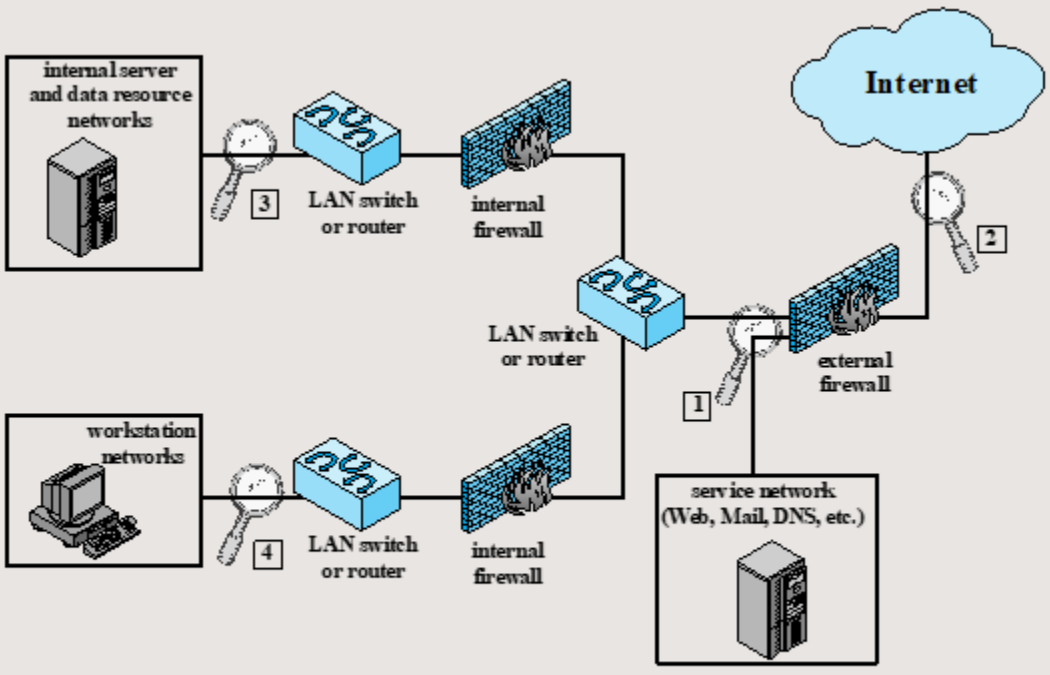


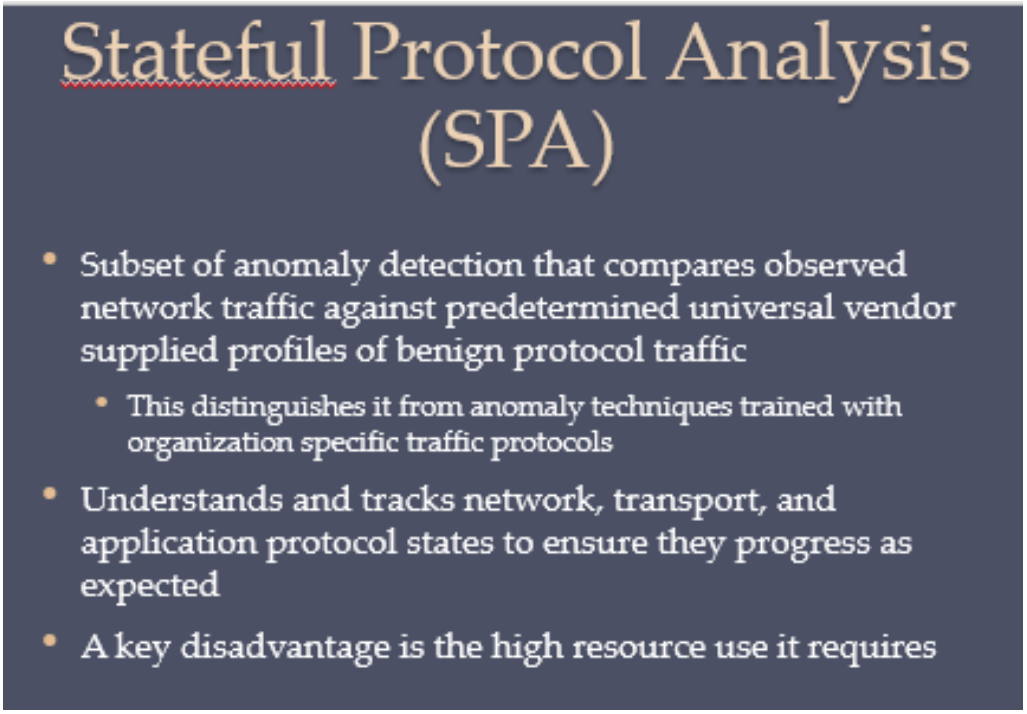
Figure 8.5 Example of NIDS Sensor Deployment

Types of attacks that are suitable for signature detection:

- **Application layer reconnaissance and attacks:** NIDS technologies analyze several dozen application protocols such as DHCP, FTP, HTTP and many more. The NIDS is looking for attack patterns that have been identified as targeting these protocols. Examples of attack include buffer overflows, password guessing, and malware transmission.
- **Transport layer reconnaissance and attacks:** NIDSs analyze TCP and UDP traffic and perhaps other transport layer protocols. Examples of attacks are unusual packet fragmentation, scans for vulnerable ports, and TCP-specific attacks such as SYN floods
- **Network layer reconnaissance and attacks**: NIDSs typically analyze IPv4, IPv6, ICMP, and IGMP at this level. Examples of attacks are spoofed IP addresses and illegal IP header values.
- **Unexpected application services:** The NIDS attempts to determine if the activity on a transport connection is consistent with the expected application protocol. An example is a host running an unauthorized application service.
- **Policy violations:** Examples include use of inappropriate Web sites and use of forbidden application protocols.

Types of attacks that are suitable for anomaly detection:

- **Denial-of-service (DoS) attacks**: Such attacks involve either significantly increased packet traffic or significantly increase connection attempts, in an attempt to overwhelm the target system. Anomaly detection is well suited to such attacks.
- **Scanning**: A scanning attack occurs when an attacker probes a target network or system by sending different kinds of packets. Using the responses received from the target, the attacker can learn many of the system's characteristics and vulnerabilities. Thus, a scanning attack acts as a target identification tool for an attacker. Scanning can be detected by atypical flow patterns at the application layer (e.g., banner grabbing3), transport layer (e.g., TCP and UDP port scanning), and network layer (e.g., ICMP scanning).
- **Worms**: Worms spreading among hosts can be detected in more than one way. Some worms propagate quickly and use large amounts of bandwidth. Worms can also be detected because they can cause hosts to communicate with each other that typically do not, and they can also cause hosts to use ports that they normally do not use. Many worms also perform scanning.



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

When a sensor detects a potential violation, it sends an alert and logs information related to the event. The NIDS analysis module can use this information to refine intrusion detection parameters and algorithms. The security administrator can use this information to design prevention techniques.

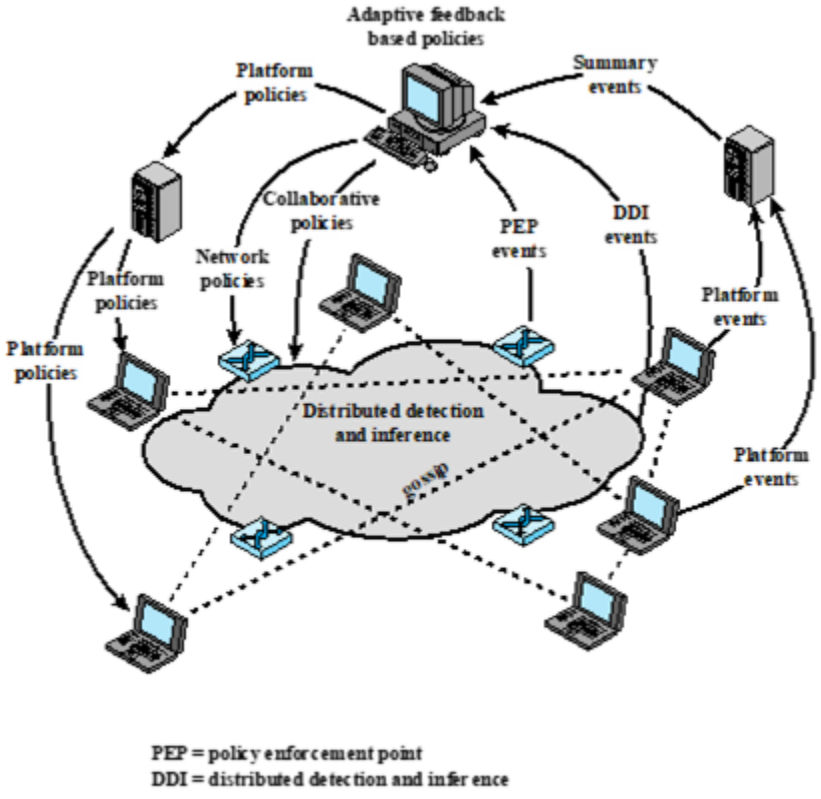


Figure 8.6 Overall Architecture of an Autonomic Enterprise Security System

In an adaptive, cooperative system, the local node instead uses a peer-to-peer “gossip” protocol to inform other machines of its suspicion, in the form of a probability that the network is under attack. If a machine receives enough of these messages so that a threshold is exceeded, the machine assumes an attack is under way and responds. The machine may respond locally to defend itself and also send an alert to a central system.