22CY701 INTRUSION DETECTION AND INTERNET SECURITY

UNIT 1 Introduction to Intrusion Detection

Course Objectives

- To Understand when, where, how, and why to apply Intrusion Detection tools and techniques in order to improve the security posture of an enterprise.
- • To Apply knowledge of the fundamentals and history of Intrusion Detection in order to avoid common pitfalls in the creation and evaluation of new Intrusion Detection Systems
- • To Analyze intrusion detection alerts and logs to distinguish attack types from false alarms
- • To Understand the fundamentals of network security, including the MAC layer, Internet Protocol, and common attacks targeting these layers.
- • To Gain an understanding of key internet security mechanisms, including firewalls, virtual private networks (VPNs), and TLS/SSL

Course Outcomes

CO1:Understand fundamental concepts and demonstrate skills in capturing and analyzing network packets.

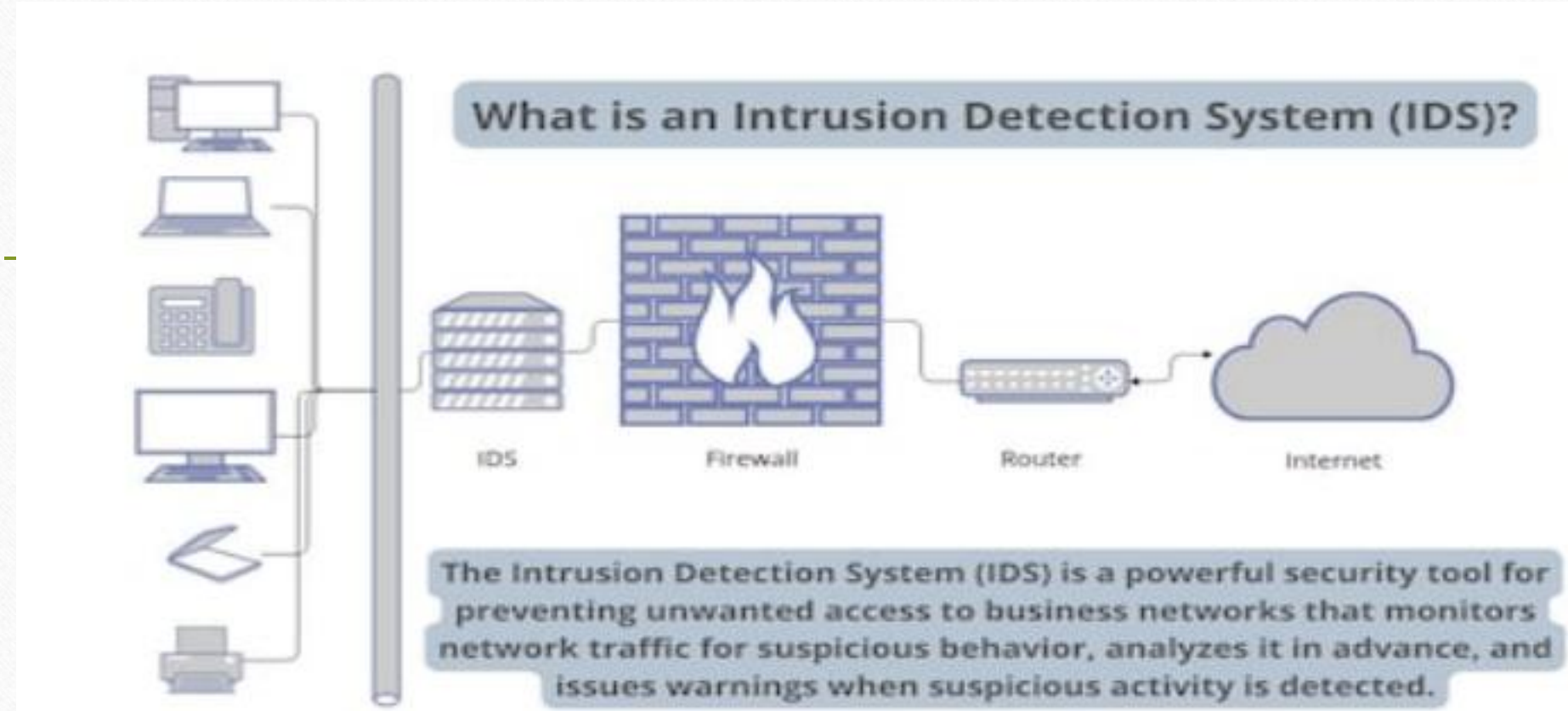
CO2: Utilize various protocol analyzers and Network Intrusion Detection Systems (NIDS) to detect network attacks and troubleshoot network problems.

CO3: Develop the ability to proficiently use the Snort tool for detecting and mitigating network attacks

CO4:Demonstrate knowledge of network security basics, including MAC layer vulnerabilities and attacks, as well as common attacks targeting the Internet Protocol.

CO5:Demonstrate understanding of firewall, VPN, and TLS/SSL VPN principles and functionalities in network security.

CO6:Apply the concepts of Intrusion Detection and internet security protocols to develop cyber security mechanisms.

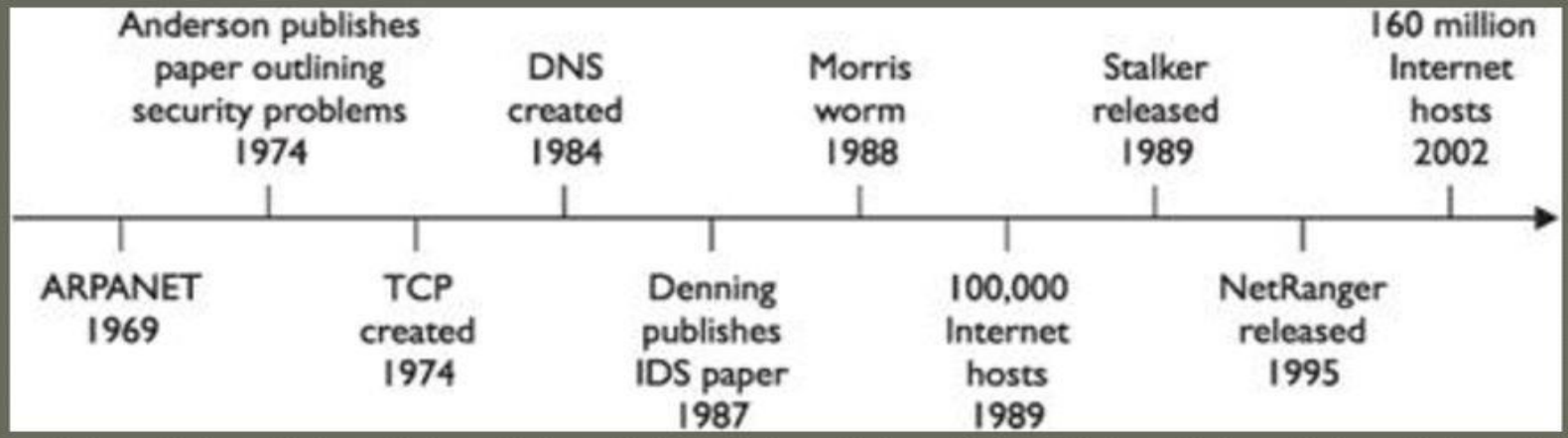


An Intrusion Detection System (IDS) is a tool that recognizes an attack on the network. It takes immediate steps to evaluate such activities and restore them to normal. Thus, IDS in security is crucial in your network. It will help you to detect traffic. IDS will immediately send an alarm.

Why IDS before Firewall

- While firewalls are the first line of defense blocking known malicious traffic, Intrusion Detection Systems (IDS) are often placed behind or integrated with firewalls to monitor for threats that may have bypassed the firewall or originate from within the network, providing an additional layer of security and visibility.

HISTORY OF IDS



About IDS

- **The goal of intrusion detection is seemingly simple: to detect intrusions.**
- they only identify evidence of intrusions, either while they're in progress or after the fact.
- Such evidence is sometimes referred to as an attack's "manifestation."
- If there is no manifestation, if the manifestation lacks sufficient information, or if the information it contains is untrustworthy, then the system cannot detect the intrusion.
- **IDSs and IPSs are just two of many methods that should be employed in a strong security program.**

About IDS

- IDSs work at the network layer of the OSI model, and passive network sensors are typically positioned at choke points on the network.
- They analyse packets to find specific patterns in network traffic—if they find such a pattern in the traffic, an alert is logged, and a response can be based on the data recorded.

About IDS

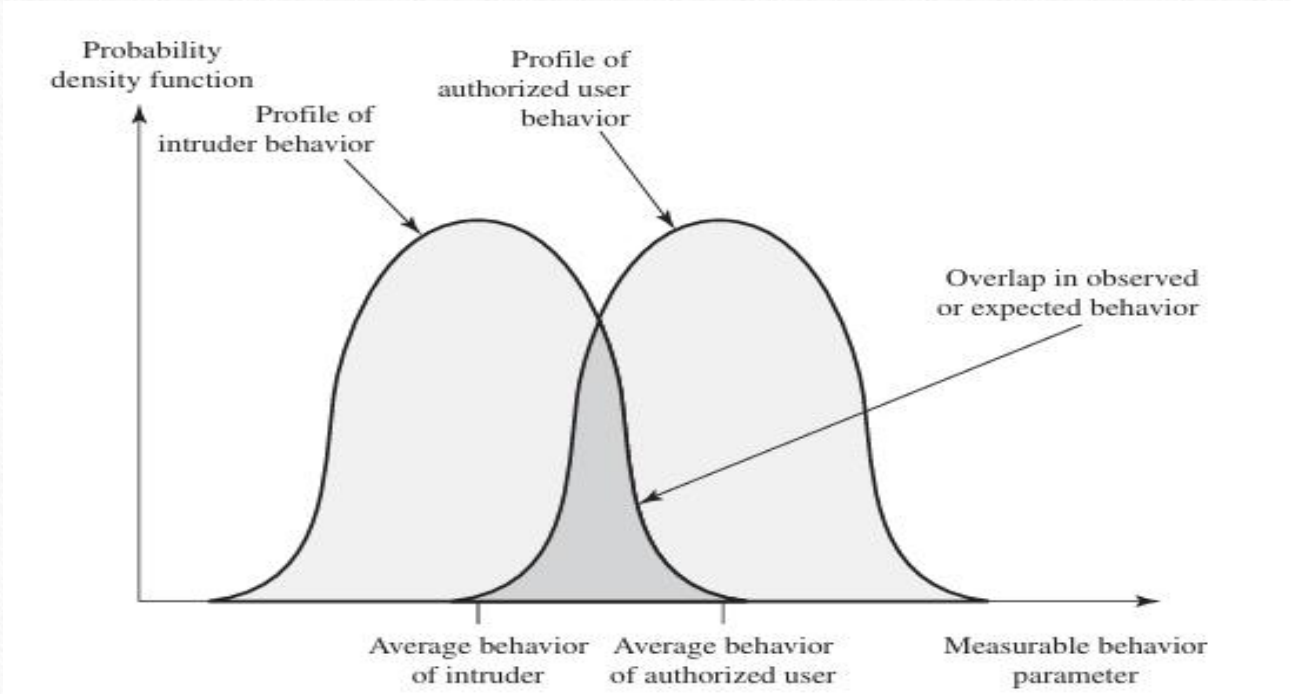
- Intrusion detection is the process of monitoring the events occurring in a computer system or network and analyzing them for signs of intrusions, **defined as attempts to compromise the confidentiality, integrity, availability,** or to bypass the security mechanisms of a computer or network.
- Intrusions are caused by attackers accessing the systems from the Internet, authorized users of the systems **who attempt to gain additional privileges for which they are not authorized, and authorized users who misuse the privileges given them.**
- Intrusion Detection Systems (IDSs) are **software or hardware products** that automate this monitoring and analysis process.

Layer	Function	Protocols
Application (user interface)	This layer is used for applications, such as HTTP, specifically written to run over the network and allows accesses to network services. It handles issues like network transparency, resource allocation, and problem partitioning. The application layer is concerned with the user's view of the network, like formatting. In addition, this layer allows access to services that support applications and handle network access, flow, and recovery.	DNS, FTP, TFTP, BOOTP, SNMP, RLOGIN, SMTP, MIME, NFS, FINGER, TELNET, APPC, AFP,
Presentation (translation)	The presentation layer helps to translate between the application and the network formats. This is also where protocol conversion takes place.	Named Pipes, Mail Slots, RPC, NCP, SMB
Session	The session layer helps to establish, maintain, and end sessions across the network.	NetBios
Transport (packets; flow control and error-handling)	The transport layer manages the flow control of data between parties across the network.	TCP, ARP, RARP, SPX, NWLink, ATP, NetBEUI
Network (addressing; routing)	The network layer translates logical network addresses and names to their physical addresses and is responsible for addressing and managing network problems such as packet switching, data congestion, and routing.	IP, ARP, RARP, ICMP, RIP, OSFP, IGMP, IPX, NWLink, OSI, DDP, DECnet
Data link (data frames to bits)	The data-link layer turns packets into raw bits on the sending end, and at the receiving end turns bits into packets. It handles data frames between the network and physical layers.	
Physical (hardware; raw bit stream)	The physical layer transmits the raw bit stream over the physical cable or airwaves (when dealing with wireless). It defines cables, cards, and other physical aspects.	IEEE 802, IEEE 802.2, ISO 2110, ISDN

Why IDS

- **If an intrusion is detected quickly enough, the intruder can be identified and ejected from the system before any damage is done or any data are compromised.**
- Even if the detection is not sufficiently timely to preempt the intruder, **the sooner that the intrusion is detected, the less the amount of damage** and the more quickly that recovery can be achieved.
- Intrusion detection is based on **the assumption that the behavior of the intruder differs from that of a legitimate user** in ways that can be quantified. Of course, we cannot expect that there will be a crisp, exact distinction between an attack by an intruder and the normal use of resources by an authorized user. Rather, we must expect that **there will be some overlap.**

Profiles of Behavior of Intruders and Authorized Users



Requirements of IDS

- **Run continually** with minimal human supervision.
- Be **fault tolerant** in the sense that it must be able to recover from system crashes and reinitializations.

- Resist subversion. The IDS **must be able to monitor itself** and detect if it has been modified by an attacker.
- Impose a **minimal overhead** on the system where it is running.
- Be **able to be configured according to the security policies** of the system that is being monitored.
- Be able to **adapt to changes in system** and user behavior over time.
- Be able to scale to **monitor a large number of hosts**.
- 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.
- Allow **dynamic reconfiguration**; that is, the ability to reconfigure the IDS without having to restart it.

IDS Architecture

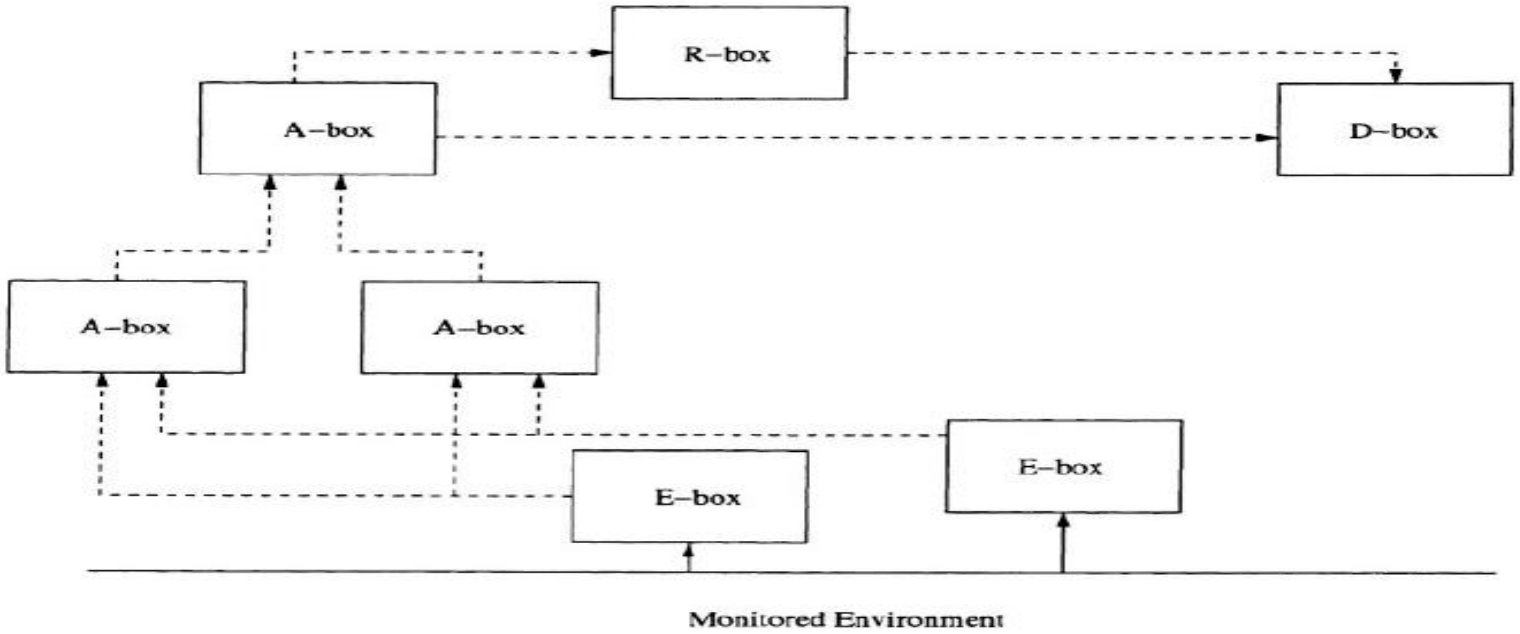


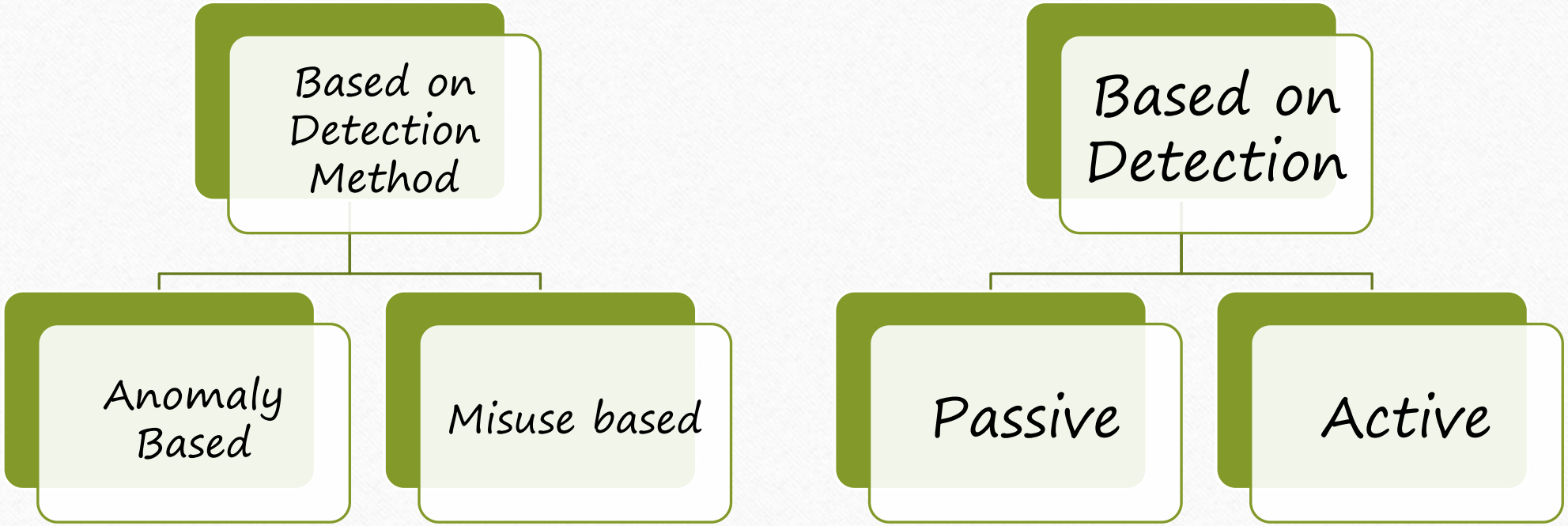
Figure 2.3. CIDF Description of an IDS System

- ***Event boxes (E-boxes).*** The role of event boxes is to generate events by processing raw audit data produced by the computational environment. A common example of an E-boxes is a program that filters audit data generated by an operating system
- ***Analysis boxes (A-boxes).*** The role of an analysis box is to analyze the events provided by other components. The results of the analysis are sent back to the system as additional events, typically representing alarms. Usually A-boxes analyze simple events supplied by E-boxes. Some A-boxes analyze events produced by other A-boxes and operate at a higher level of abstraction.

Database boxes (D-boxes). Database boxes simply store events, guaranteeing persistence and allowing postmortem analysis.

Response boxes (R-boxes). Response boxes consume messages that carry directives about actions to be performed as a reaction to a detected intrusion. Typical actions include killing processes, resetting network connections, and modifying firewall settings.

Taxonamy


Based on Detection Method
Anomaly Based
Misuse based
Based on Detection
Passive
Active

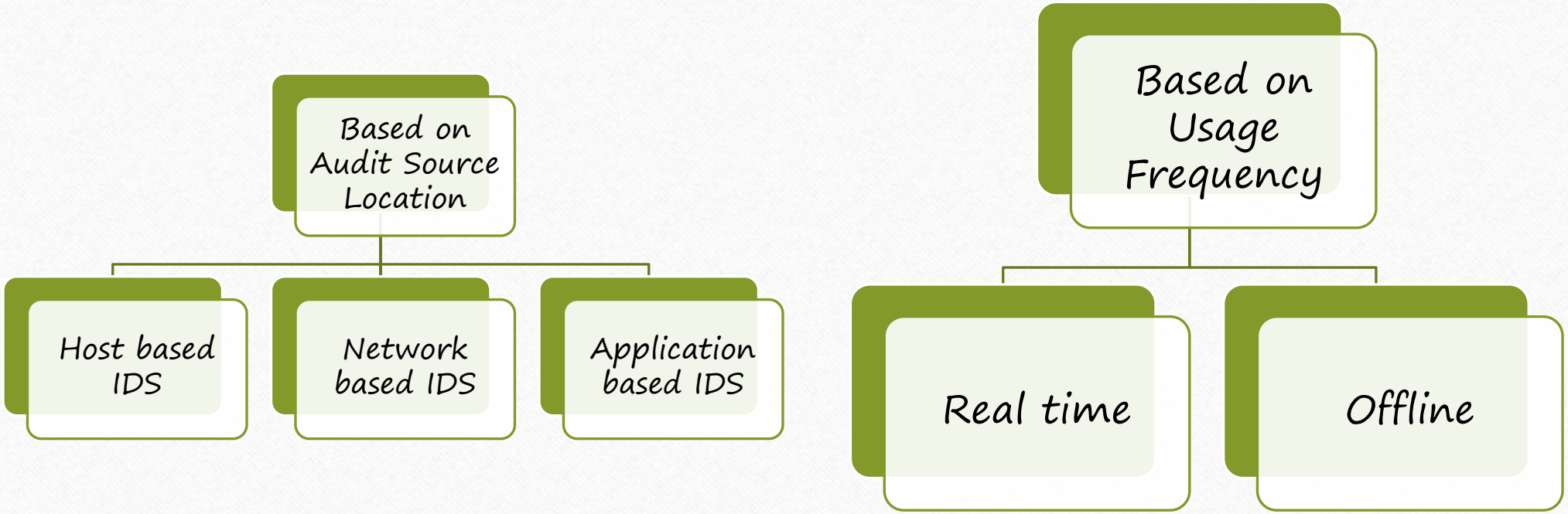
Taxonomy

- ***Detection method.*** It defines the philosophy on which the **A-box is built**. Two approaches have been proposed.
- When the IDS defines what is "normal" in the environment and **flags as attacks deviations from normality**, it is qualified as ***anomaly-based.***
- When the IDS explicitly defines what is "abnormal", **using specific knowledge about the attacks** in order to detect them, it is called ***misuse-based.***

Taxonomy

- ***Behavior on detection.*** It defines a **characteristic of the R-box.**
- It is said to be **passive if the system just issues an alert when an attack is detected.**
- If **more proactive actions are taken** (e.g., disconnecting users, shutting down network connections), it is said to be active.

Taxonomy


Based on Audit Source Location
Host based IDS
Network based IDS
Application based IDS
Based on Usage Frequency
Real time
Offline

Taxonomy

- ***Audit source location.*** It specifies where the **E-box** takes audit data from. We distinguish between **host-based IDSs**, which deal with audit data generated on a single host, e.g., a C2 audit trail; **application-based IDSs,** which work on audit records produced by a specific application; and **network-based IDSs**, which monitor network traffic.
- ***Usage frequency.*** It discriminates between systems that **analyze the data in real time** and those that are run periodically (offline). It specifies how often the **A-box** analyzes data collected by other parts of the system.

IDS Audit

- An **IDS Audit** refers to the **process of systematically reviewing and analyzing the logs, alerts, and data generated by an Intrusion Detection System (IDS)** to assess its effectiveness, accuracy, and compliance with security policies.

Purpose of IDS Audit

- **Verify IDS Effectiveness:**
  To check if the IDS correctly identifies real attacks or malicious activities on the network or host systems.
- **Reduce False Positives and False Negatives:**
  False positives are benign activities mistakenly flagged as intrusions, while **false negatives are actual intrusions that the IDS misses**. An audit helps tune the IDS rules and algorithms to minimize these.
- **Compliance and Accountability:**
  Many industries and organizations have regulatory or policy requirements that mandate regular security audits, including IDS audits.

📊 Confusion Matrix Elements

	Predicted: Attack	Predicted: Normal
Actual: Attack	✅ True Positive (TP)	❌ False Negative (FN)
Actual: Normal	❌ False Positive (FP)	✅ True Negative (TN)

Purpose of IDS Audit

- **Incident Investigation:**
  IDS audit logs are crucial during forensic investigations after a security incident to understand the attack vector, timeline, and impact.
- **Performance Improvement:**
  Helps in identifying performance bottlenecks or gaps in IDS coverage.

Components of IDS audit

- **Review of IDS Logs and Alerts:**
  - Analyze recorded intrusion events.
  - Check patterns and frequency of alerts.
  - Validate whether alerts correspond to real security events.
- **Evaluation of IDS Configuration:**
  - Verify that IDS sensors are properly placed in the network for optimal coverage.
  - Check detection rules and signatures for relevance and currency.
  - Assess whether the IDS is updated regularly with latest threat signatures.
- **Assessment of IDS Response:**
  - Check how alerts are handled — are they escalated properly?
  - Evaluate the workflow for incident response triggered by IDS alerts.

Components of IDS audit

- **Analysis of False Positives/Negatives:**
  - Identify events that were wrongly classified.
  - Understand the reasons and update IDS signatures or tuning parameters accordingly.
- **System Performance Review:**
  - Examine if the IDS can handle traffic loads without dropping packets.
  - Assess resource utilization (CPU, memory).
- **Compliance Verification:**
  - Ensure audit trails meet organizational or regulatory standards.
  - Check retention periods and access controls for IDS logs.

Components of IDS audit

- **Manual Audit:**
  Security analysts manually review IDS logs and reports.
- **Automated Audit:**
  Using audit tools and software that parse IDS data, detect anomalies, and generate audit reports.
- **Hybrid Audit:**
  Combines automated tools with human review for greater accuracy.

Tools and Techniques Used in IDS Audit

- **Log Analysis Tools:**
  Tools like **Splunk, ELK Stack** (Elasticsearch, Logstash, Kibana), or proprietary **SIEM (Security Information and Event Management)** systems to aggregate and analyze IDS data.
- **Statistical Analysis:**
  Identifying trends and patterns in IDS alerts to detect abnormal behavior.
- **Signature and Rule Review:**
  Regularly checking the IDS detection rules for outdated or missing signatures.
- **Simulation Testing:**
  Running penetration tests or simulated attacks to check if IDS detects them.

Challenges in IDS Audit

- High volume of data and alerts to analyze.
- Distinguishing between real attacks and benign anomalies.
- Keeping IDS signatures and rules updated.
- Balancing detection sensitivity to avoid excessive false alarms.
- Skilled personnel required to interpret audit findings correctly.

SNORT Installation

https://letsdefend.io/blog/how-to-install-and-configure-snort-on-ubuntu

Threats

Category	Description	Example
Internal Threat	Comes from inside the organization	Employee stealing data
External Threat	Comes from outside the organization	Hacker phishing attack
Structured Attack	Well-planned and targeted	Nation-state ransomware
Unstructured Attack	Random and unskilled	Amateur trying DDoS tools

Intruders

- **Masquerader:** An individual **who is not authorized** to use the computer and who penetrates a system's access controls to exploit a legitimate user's account
- **Misfeasor:** A legitimate user who accesses data, programs, or resources for which such access is not authorized, or who is **authorized for such access but misuses his or her privileges**
- **Clandestine user:** An individual who **seizes supervisory control of the system and uses this control to evade auditing and access controls or to suppress audit collection**

Examples of Intrusions

- Performing a remote root compromise of an e-mail server
- Defacing a Web server
- Guessing and cracking passwords
- Copying a database containing credit card numbers
- Viewing sensitive data, including payroll records and medical information, without authorization
- Running a packet sniffer on a workstation to capture usernames and passwords
- Using a permission error on an anonymous FTP server to distribute pirated software and music files
- Dialling into an unsecured modem and gaining internal network access
- Posing as an executive, calling the help desk, resetting the executive's e-mail password, and learning the new password
- Using an unattended, logged-in workstation without permission

Intruder Behavior Patterns

(a) Hacker

1. Select the target using IP lookup tools such as NSLookup, Dig, and others.
2. Map network for accessible services using tools such as NMAP.
3. Identify potentially vulnerable services (in this case, pcAnywhere).
4. Brute force (guess) pcAnywhere password.
5. Install remote administration tool called DameWare.
6. Wait for administrator to log on and capture his password.
7. Use that password to access remainder of network.

(b) Criminal Enterprise

1. Act quickly and precisely to make their activities harder to detect.
2. Exploit perimeter through vulnerable ports.
3. Use Trojan horses (hidden software) to leave back doors for reentry.
4. Use sniffers to capture passwords.
5. Do not stick around until noticed.
6. Make few or no mistakes.

(c) Internal Threat

1. Create network accounts for themselves and their friends.
2. Access accounts and applications they wouldn't normally use for their daily jobs.
3. E-mail former and prospective employers.
4. Conduct furtive instant-messaging chats.
5. Visit Web sites that cater to disgruntled employees, such as f'dcompany.com.
6. Perform large downloads and file copying.
7. Access the network during off hours.

Need for IDS

- Detects Unauthorized Access
- Provides Early Alerts
- Monitors Network and Host Behavior
- Complements Other Security Tools
- Detects Insider Threats
- Supports Incident Response
- Enables Forensic Analysis
- Assists in Regulatory Compliance
- Reduces Downtime and Data Loss
- Detects Known and Unknown Attacks
- Preventing problems by increasing the perceived risk of discovery and punishment of attackers
- Detecting problems that are not prevented by other security measures
- Detecting the preambles to attacks (often experienced as network probes and other tests for existing vulnerabilities)
- Documenting the existing threat
- Quality control for security design and administration
- Providing useful information about actual intrusions

Process model for Intrusion Detection

- ***Information Sources*** – the different sources of event information used to determine whether an intrusion has taken place. These sources can be drawn from different levels of the system, with network, host, and application monitoring most common.
- ***Analysis*** – the part of intrusion detection systems that actually organizes and makes sense of the events derived from the information sources, deciding when those events indicate that intrusions are occurring or have already taken place. The most common analysis approaches are *misuse detection* and *anomaly detection*.
- ***Response*** – the set of actions that the system takes once it detects intrusions. These are typically grouped into active and passive measures, with active measures involving some automated intervention on the part of the system, and passive measures involving reporting IDS findings to humans, who are then expected to take action based on those reports.

Host-Target Co-location

- In early days of IDSs, most IDSs ran on the systems they protected. This was due to the fact that most systems were mainframe systems, and the cost of computers made a separate IDS system a costly extravagance.
- This presented a problem from a **security point of view, as any attacker that successfully attacked the target system could simply disable the IDS as an integral portion of the attack.**

Host-Target Separation

- With the advent of workstations and personal computers, **most IDS architects moved towards running the IDS control and analysis systems on a separate system, hence separating the IDS host and target systems.**
- This improved the security of the IDS as this made it much easier to hide the existence of the IDS from attackers.

Goals of IDS

- *Accountability:*
- *"I can deal with security attacks that occur on my systems as long as I know who did it (and where to find them.)"*
- Accountability is difficult in TCP/IP networks, where the protocols allow attackers to forge the identity of source addresses or other source identifiers.

Goals of IDS

- Response

"I don't care who attacks my system as long as I can recognize that the attack is taking place and block it."

Goals of IDS

Control Strategy

Control Strategy describes how the elements of an IDS is controlled, and furthermore, how the input and output of the IDS is managed.

Centralized

- Under centralized control strategies, **all monitoring, detection and reporting is controlled directly from a central location**

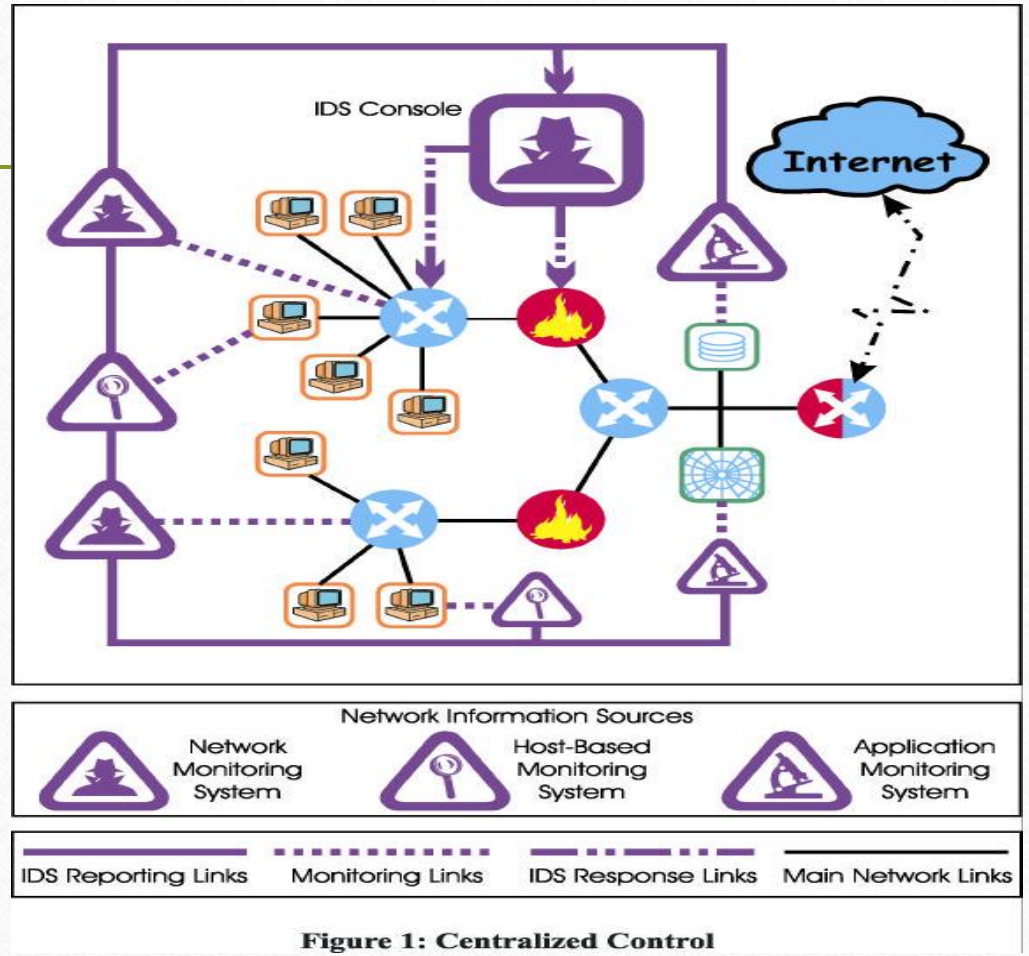


Figure 1: Centralized Control

Partially Distributed

- **Monitoring and detection is controlled from a local control node**, with hierarchical reporting to one or more central location(s).

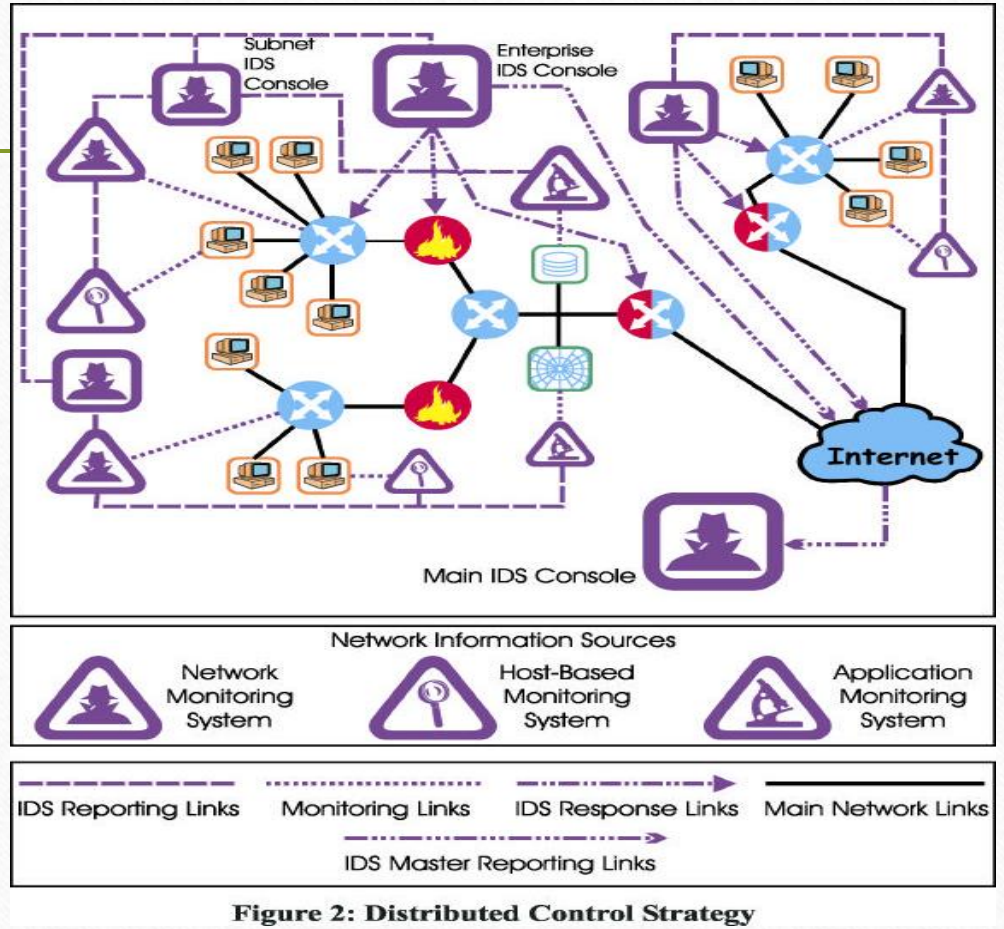


Figure 2: Distributed Control Strategy

Fully Distributed

- **Monitoring and detection is done using an agent-based approach**, where response decisions are made at the point of analysis.

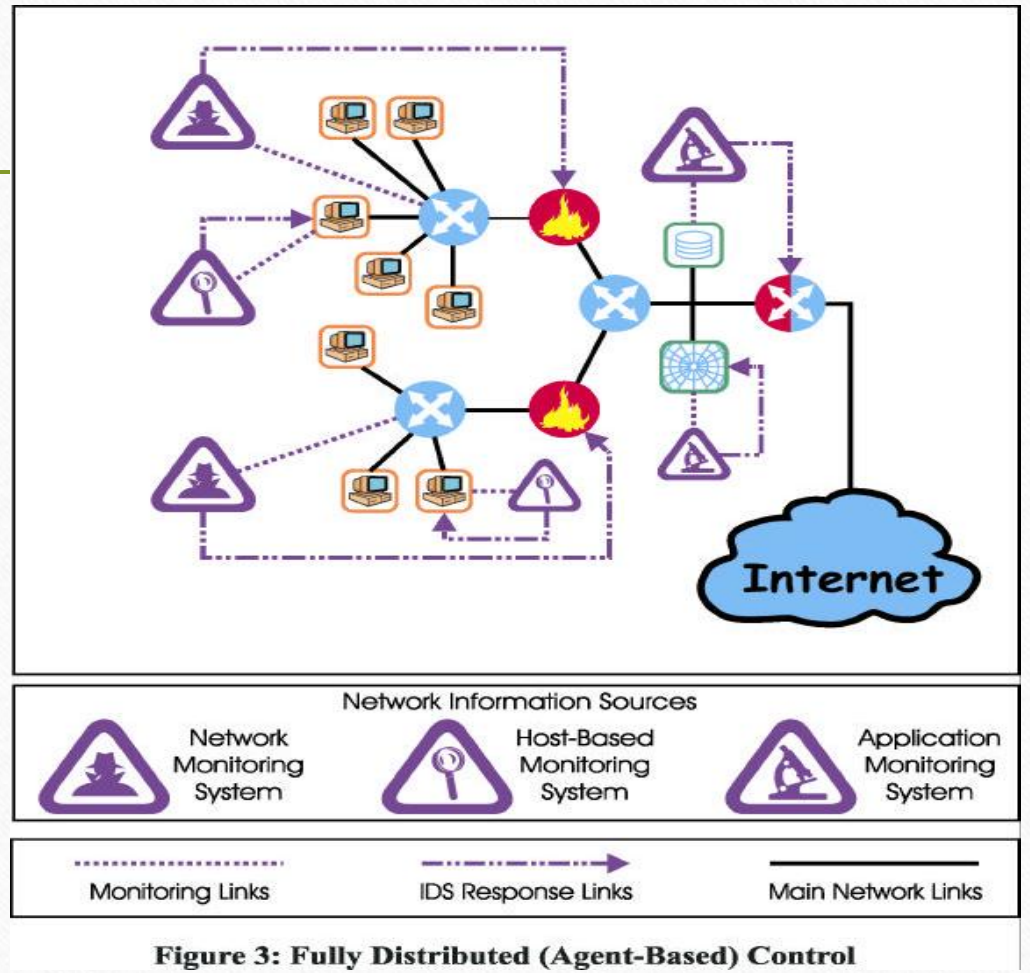


Figure 3: Fully Distributed (Agent-Based) Control

Timing

- **Timing refers to the elapsed time between the events that are monitored and the analysis of those events.**
- **Interval-Based (Batch Mode)**

In interval-based IDSs, the information flow from monitoring points to analysis engines is not continuous. In effect, the information is handled in a fashion similar to **"store and forward"** communications schemes.

Many early host-based IDSs used this timing scheme, as they relied on operating system audit trails, which were generated as files. Interval based IDSs are precluded from performing active responses.

Timing

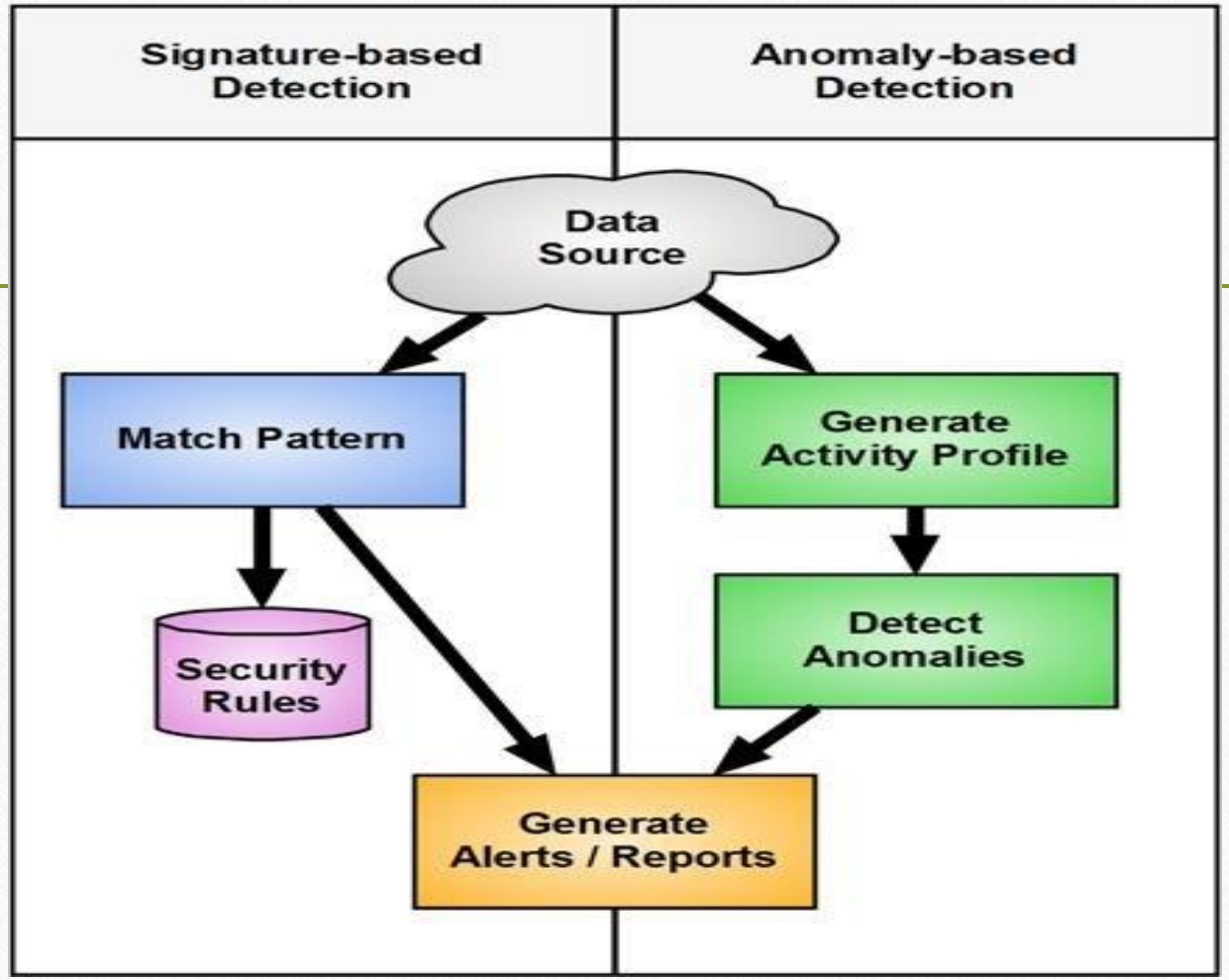
- **Real-Time(Continuous)**
- Real-time IDSs operate on continuous information feeds from information sources. This is the predominant timing scheme for network based IDSs, which gather information from network traffic streams.
- This means that **detection performed by a "real-time" IDS yields results quickly enough to allow the IDS to take action that affects the progress of the detected attack.**

IDS Techniques

Rule-Based Detection(Misuse Detection)

- Misuse detectors analyze system activity, looking for events or sets of events that **match a predefined pattern of events** that describe a **known attack.**
- The four phases of the analysis process are applied in a rule-based detection system:

1. Preprocessing The first step is to **collect data about intrusions, vulnerabilities, and attacks, and put them into a classification scheme or pattern descriptor**.


Signature-based Detection
Anomaly-based Detection
Data Source
Match Pattern
Security Rules
Generate Activity Profile
Detect Anomalies
Generate Alerts / Reports

Rule-Based Detection(Misuse Detection)

PATTERN DESCRIPTORS

- **Signature Name** The given name of a signature
- **Signature ID** A unique ID for the signature
- **Signature Description** Description of the signature and what it does
- **Possible False Positive Description** An explanation of any "false positives" that may appear to be an exploit but are actually normal network activity.
- **Related Vulnerability Information** This field has any related vulnerability information
- **User Notes** This field allows a security professional to add specific notes related to their network

Rule-Based Detection(Misuse Detection)

- The pattern descriptors are typically either **content-based signatures**, which examine the **payload and header of a packet**, or **context-based signatures** that evaluate ***only*** **the packet headers** to identify an alert.
- An **atomic descriptor requires only one packet to be inspected to identify an alert**, while a **composite descriptor requires multiple packets to be inspected to identify an alert.**
- The pattern descriptors are then put into a knowledge base that contains the criteria for analysis.

Rule-Based Detection(Misuse Detection)

2. Analysis The event data are formatted and **compared against the knowledge base** by using a pattern-matching analysis engine. The analysis engine looks for defined patterns that are known as attacks.

3. Response

- If the event **matches the pattern of an attack**, the analysis engine sends an alert.
- If the event is a **partial match, the next event is examined**.
- Note that **partial matches can only be analyzed with a stateful detector**, which has the ability to maintain state, as many IDS systems do.
- Different responses. can be returned depending on the specific event records.

Rule-Based Detection(Misuse Detection)

4. Refinement

Refinement of pattern-matching analysis **comes down to updating signatures**, because an IDS is only as good as its latest signature update. This is one of the drawbacks of pattern-matching analysis. Most IDSs allow **automatic and manual updating of attack signatures.**

Rule-Based Detection(Misuse Detection)

- *Advantages*

- Very effective at detecting attacks **without generating an overwhelming number of false alarms.**
- **Quickly and reliably diagnose the use of a specific attack tool or technique.** This can help security managers prioritize corrective measures.
- Allow system managers, **regardless of their level of security expertise, to track security problems on their systems, initiating incident handling procedures.**

- *Disadvantages*

- Misuse detectors can only detect those attacks they know about – therefore they must be **constantly updated with signatures of new attacks.**
- Many misuse detectors are designed to use **tightly defined signatures that prevent them from detecting variants of common attacks.**

Profile Based Detection (Anomaly Detection)

- Anomaly detectors **identify abnormal unusual behavior (anomalies) on a host or network.**
- They function on the **assumption that attacks are different from "normal" (legitimate) activity** and can therefore be detected by systems that identify these differences.
- Anomaly detectors **construct profiles representing normal behavior of users, hosts, or network connections.**
- These **profiles are constructed from historical data collected over a period of normal operation.**
- The detectors then collect event data and use a variety of measures to determine when monitored activity deviates from the norm.

Profile Based Detection (Anomaly Detection)-Threshold detection

- Threshold detection, in which certain attributes of user and system behavior are expressed in terms of counts, with some level established as permissible.
- Such behavior

❑ **attributes can include the number of files accessed by a user in a given period of time,**

❑ **the number of failed attempts to login to the system,**

❑ **the amount of CPU utilized by a process, etc.**

- This level can be **static or heuristic** (i.e., designed to change with actual values observed over time)

→ **used in commercial IDS**

Profile Based Detection (Anomaly Detection)- Statistical measures

- Statistical measures, both parametric, where the distribution of the profiled attributes is assumed to fit a particular pattern, and non-parametric, where the distribution of the profiled attributes is "learned" from a set of historical values, observed over time.

→ used in commercial IDS

📌 Example: Monitoring Network Traffic Volume

1. Parametric Statistical Measure

- **Assumption:** Network traffic follows a **Normal (Gaussian) distribution**.
- **Profile:** Average number of packets per minute during normal operation = **100 packets/min**, with a standard deviation (σ) = **15**.
- **Threshold Rule:** Any value beyond **μ ± 3σ (55 to 145 packets/min)** is treated as an anomaly.

Scenario

- At 2:00 PM, the IDS observes **180 packets/min**.
- Since 180 > 145, the IDS flags this as an **anomaly** → possible **DDoS attack**.

(Here, the parametric model depends on the assumption of a normal distribution.)

2. Non-Parametric Statistical Measure

- **Assumption:** No specific distribution is assumed; instead, the IDS uses **historical data** to learn what's normal.
- **Profile:** From the last 30 days of logs, the IDS sees that **95% of packet counts per minute** lie between **70 and 130**.
- **Threshold Rule:** Any observation outside this historical range is suspicious.

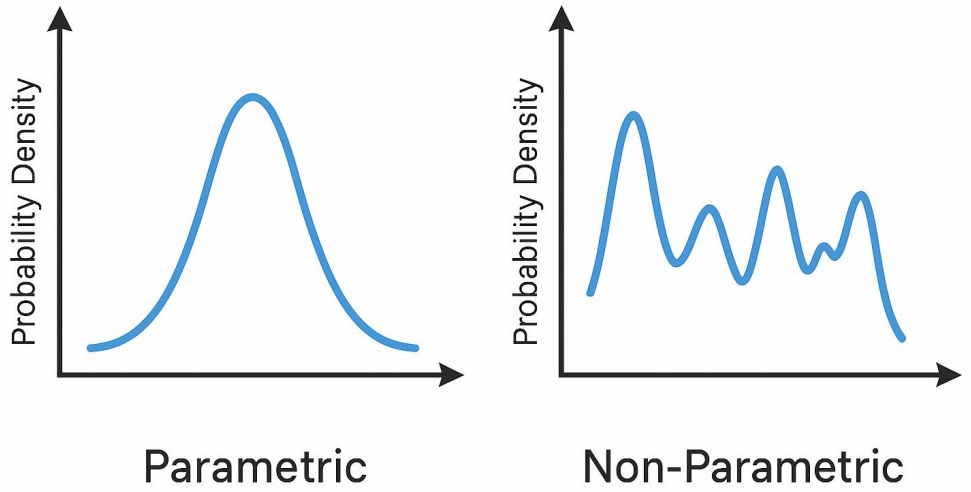
Scenario

- At 10:30 AM, the IDS sees **150 packets/min**.
- Since 150 is outside the learned 95% range, the IDS raises an alert.

(Here, the model adapts based on past data instead of assuming a theoretical pattern.)

Parametric vs Non Parametric

- Parametric: distribution assumed to fit a particular pattern
- Non-parametric: distribution “learned” from historical values



Profile Based Detection (Anomaly Detection)- Rule-based measures

- Rule-based measures, which are **similar to non-parametric statistical measures** in that observed data **defines acceptable usage patterns**, but differs in that those patterns are specified as rules, not numeric quantities

Rule-Based Measure

- Instead of numbers, the administrator sets **explicit rules** like:
  - *A user should not access files in the "Payroll" folder unless they are in the HR group.*
  - *A user must not download more than 3 confidential files within 10 minutes.*
  - *No process should attempt to read* `/etc/shadow` *(Linux password file).*

Scenario

- A marketing employee attempts to open a file in the Payroll folder.
- The IDS checks the rule: *"Only HR users can access Payroll."*
- Since the employee is not in HR, the IDS flags this as an **anomaly**.

Profile Based Detection (Anomaly Detection)-Analysis

- ***Behavioral analysis*** looks for anomalies in the types of behavior that have been statistically baselined, such as relationships in packets and what is being sent over a network.
- ***Traffic-pattern analysis*** looks for specific patterns in network traffic.
- ***Protocol analysis*** looks for network protocol violations or misuse based on RFC-based behavior.

Analysis model of Anomaly Detection

1. **Preprocessing** The first step in the analysis process is **collecting the data in which behavior considered normal on the network is baselined over a period of time**. The data are **put into a numeric form and is then formatted**. Then the information is classified into a **statistical profile that is based on different algorithms** in the knowledge base.
2. **Analysis** The event data are typically reduced to a **profile vector**, which is then compared to the knowledge base. **The contents of the profile vector are compared to a historical record for that particular user, and any data that fall outside of the baseline normal activity is labeled a deviation.**

Profile Vector

A **profile vector** is a structured set of numerical or categorical features that represent a **user's or system's behavior** during a specific time window.

- It condenses raw event data (like logins, file accesses, CPU usage) into a feature vector.
- The IDS then compares this vector against a **knowledge base** or **historical baseline** of normal activity.
- If the profile vector deviates significantly, the event is flagged as an **anomaly**.

🔹 **Example: User Login Behavior Profile Vector**

Suppose we are monitoring user *Mr. Ravi* on a corporate system.
Over a 1-hour interval, the IDS collects the following features:

Feature	Value
Number of login attempts	4
Number of failed logins	1
Number of files accessed	25
Number of emails sent	5
CPU usage (%)	35

➡️ This set of values is the **profile vector** for Mr. Ravi in that hour:

Profile Vector = [4. 1. 25. 5. 35]

Analysis model of Anomaly Detection

3. **Response** At this point, a response can be **triggered either automatically or manually.**

4. **Refinement** The data records must be kept updated. The **profile vector history will typically be deleted after a specific number of days.** In addition, **different weighting systems can be used to add more weight to recent behaviors than past behaviors.**

Profile Based Detection (Anomaly Detection)

- *Advantages:*
- Detect unusual behavior and thus have the ability to **detect symptoms of attacks without specific knowledge of details.**
- Produce information that can in turn be **used to define signatures for misuse detectors.**

- *Disadvantages:*
- Produce a **large number of false alarms** due to the unpredictable behaviors of users and networks.
- **Require extensive "training sets"** of system event records in order to characterize normal behavior patterns.

Information Sources

Information Sources

- To detect and respond to suspicious activity, an IDS gathers data from multiple **information sources**, which act as the **eyes and ears of the system**.
- These sources provide critical input about **system behavior, network traffic, application usage, and security events.**
- Based on this information, the IDS analyzes activity to identify potential intrusions, policy violations, or malicious actions.

Host-Based Information Sources

- **System logs** – Records of user logins, command history, or system events.
- **Audit trails** – Tracks user activities, such as file access or administrative actions.
- **File integrity checking tools** – Detects unauthorized changes to system or config files (e.g., Tripwire).
- **Process monitoring** – Observes running processes and memory usage.
- **Registry changes** – On Windows systems, changes to the registry can indicate malware activity

Example:
If a critical system file is modified without authorization, the HIDS detects and flags the change.

📌 Example Scenario: Detecting Unauthorized File Access on a Web Server

Situation:
A company's web server runs a **host-based IDS (HIDS)** such as **OSSEC** or **Wazuh**. An attacker gains a normal user login and tries to read a restricted file `/etc/shadow` (which contains hashed passwords).

The HIDS uses **host-based information sources** like:

- **System logs** (auth logs, syslog, application logs)
- **File integrity checks** (changes to sensitive files)
- **System call monitoring** (tracking read/write attempts)
- **User login activity** (failed and successful logins)

How Detection Happens

1. **Login Event Captured**
   - The attacker logs in as a normal user (recorded in `/var/log/auth.log`).
2. **System Call Monitoring**
   - The user runs `cat /etc/shadow`.
   - The HIDS monitors this **read attempt**.
3. **Policy Violation Check**
   - Rule: "Normal users should not access `/etc/shadow`."
   - The HIDS matches the attempt against this policy.
4. **Alert Generation**
   - The HIDS raises an alert with details: user, file accessed, timestamp.
5. **Admin Notification**
   - Security admin is notified to investigate and block the attacker.

Network-Based Information Sources

- These sources capture and analyze data from the **network layer**. A **Network-Based IDS (NIDS)** monitors real-time network traffic and communication between systems.

Network-Based Information Sources

Common sources include:

- **Packet captures (PCAP)** – Full or partial snapshots of network traffic for in-depth inspection.
- **Firewall logs** – Shows allowed or blocked traffic, source and destination IPs, and port access.
- **Router and switch logs** – Records routing behavior and traffic patterns.
- **NetFlow or sFlow data** – Summarized traffic statistics from routers.
- **DNS logs** – Helps detect domain spoofing or exfiltration via DNS tunnels.

Example:
If a large number of failed login attempts are detected from a single IP, the NIDS can raise an alert for a brute-force attack.

✅ Typical NetFlow Parameters Collected

- Source IP & Port
- Destination IP & Port
- Protocol (TCP/UDP/ICMP)
- Number of packets
- Number of bytes
- Flow start & end time
- TCP flags (if TCP)
- Type of Service (ToS)

📌 Example Scenario: Detecting a Port Scan Attack

Situation:

An organization's IDS is monitoring its corporate network traffic at the gateway router. An attacker tries to perform a **port scan** on the organization's server (to find open ports for later exploitation).

The IDS uses **network-based information sources** such as:

- **Packet headers** (IP addresses, ports, protocols)
- **Payload analysis** (signature of suspicious content)
- **Traffic flow data** (number of connection attempts, frequency, etc.)

How the IDS Works Here:

1. **Packet Capture:** IDS sensor (e.g., Snort) collects raw packets from the network link.
2. **Feature Extraction:** It extracts fields like source IP, destination IP, ports, TCP flags, etc.
3. **Detection Logic:**
   - It notices a single external IP making **multiple connection attempts** to different ports on the same internal server in a short time.
   - This matches a **signature/rule for port scanning**.
4. **Alert Generation:** IDS raises an alert and logs the event.
5. **Action:** The security admin can check the alert and, if needed, block the IP using a firewall.

Application-Based Information Sources

- These sources are specific to **software applications** and services running on a system.
- An **Application Protocol-Based IDS (APIDS)** focuses on detecting abnormal behavior in application-level protocols.

Application-Based Information Sources

Common sources include:

- **Web server logs (Apache, Nginx)** – Tracks page visits, errors, and request patterns.
- **Database access logs** – Records queries and transactions.
- **Email server logs** – Useful for detecting spam, phishing, or malware-laden emails.
- **Authentication logs** – Records user login attempts, failures, and session durations.

📌 Example Scenario: Detecting SQL Injection via Application-Based IDS

Situation:

A university's online exam portal runs a **Web Application IDS (WAF/APPIDS)** such as **ModSecurity**.

An attacker tries to bypass login by entering a malicious SQL payload in the username field:

matlab Copy Edit

```
' OR '1'='1'; --
```

The IDS uses **application-based information sources** such as:

- Application logs (web server logs, error logs)
- HTTP request contents (POST/GET parameters, cookies)
- Application-level events (login failures, input validation errors)
- Session behavior (sudden privilege escalation or anomalies in user actions)

🔎 How Detection Works

1. **Request Inspection**
   - The IDS monitors the incoming HTTP request body for suspicious patterns.
2. **Pattern Match**
   - Detects the SQL injection attempt (`' OR '1'='1`) in the **username parameter**.
3. **Policy Check**
   - Application IDS rules specify: "Block SQL injection keywords in login fields."
4. **Alert + Action**
   - The IDS raises an alert and optionally blocks the request before it reaches the backend database.

Security Device and System Logs

- Modern IDS implementations often integrate with other **security systems** to correlate events and gain broader visibility.

Security Device and System Logs

Common sources include:

- **Antivirus and EDR logs** – Detect malware infections or suspicious user behavior.
- **Firewall and VPN logs** – Monitor perimeter activity and remote access attempts.
- **SIEM (Security Information and Event Management) tools** – Aggregate logs from various systems for correlation and alerting.
- **Intrusion Prevention System (IPS) logs** – May provide dropped packet info or real-time blocking data.

Example:
A combination of VPN logs showing login from an unusual country, and antivirus logs indicating malware, could trigger a high-priority alert.