UNIT II- INTRUSION DETECTION AND PREVENTION TECHNIQUES

Intrusion Prevention Systems, Network IDs protocol based IDs , Hybrid IDs, Analysis schemes, thinking about intrusion. A model for intrusion analysis, techniques Responses requirement of responses, types of responses mapping responses to policy Vulnerability analysis, credential analysis non credential analysis

Network based IDS

Network IDS

- A **Network Intrusion Detection System (NIDS)** is a security solution that **monitors and analyzes network traffic for malicious activities or policy violations**. Unlike host-based IDS, which is installed on individual devices, NIDS is **deployed at strategic locations** within the network to examine traffic from all devices.

Network IDS

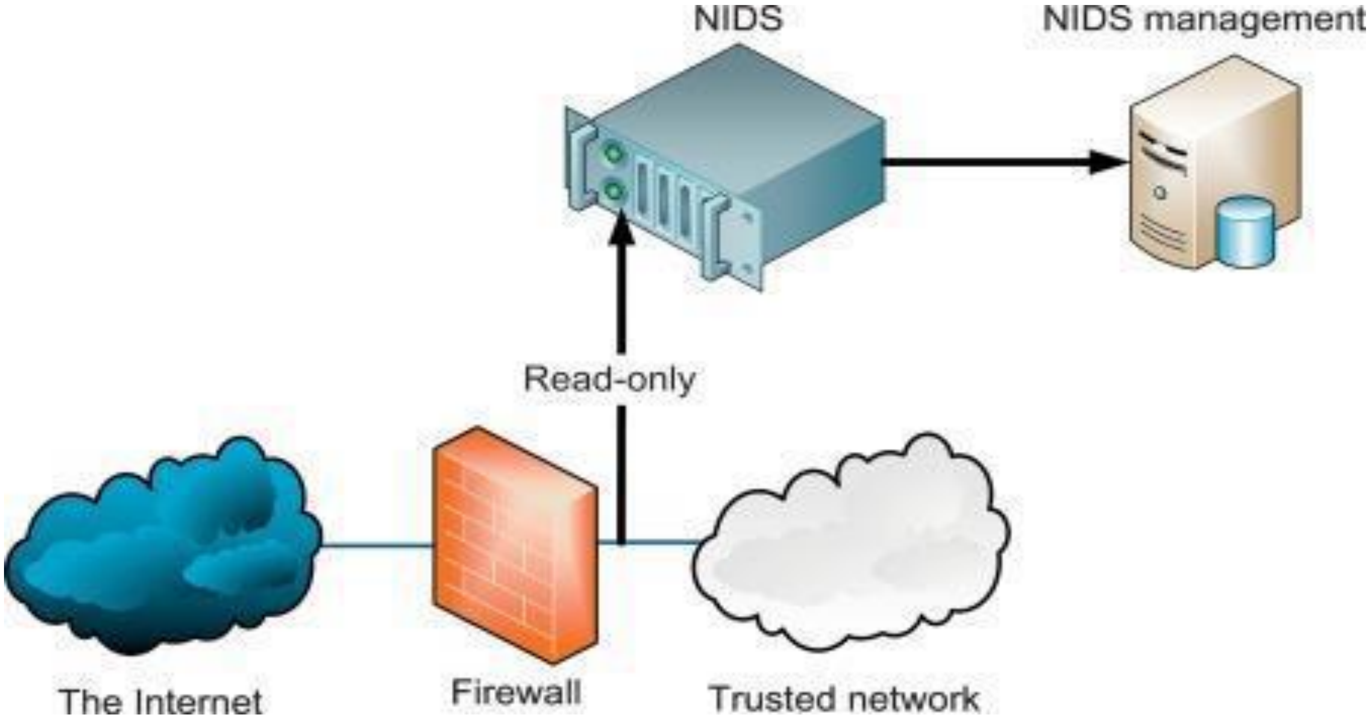
The purpose of Network IDS is to

- Detect **network-based threats like DoS attacks, port scans, malware transmissions.**
- Analyze packet-level data for suspicious patterns.
- Generate alerts for system administrators.

Components of NIDS

- **Sensors:** Capture network packets.
- **Analyzers:** Analyze the packets to detect attacks.
- **Management Console:** View alerts, logs, and reports.
- **Signature/Anomaly Database:** Contains known attack patterns or behavioral baselines.

NIDS

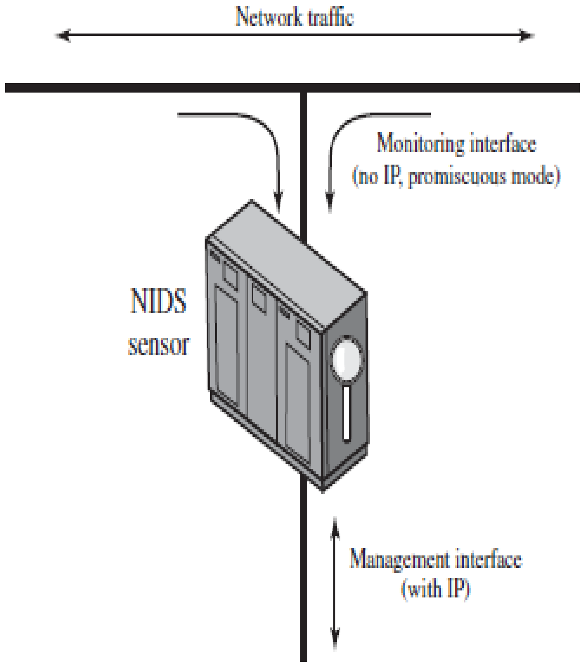

NIDS
NIDS management
Read-only
The Internet
Firewall
Trusted network

Sensors

- An **inline sensor** is inserted into a network segment so that the traffic that it is monitoring must pass through the sensor.
- One way to achieve an **inline sensor is to combine NIDS sensor logic with another network device, such as a firewall or a LAN switch.** This approach has the advantage that no additional separate hardware devices are needed; all that is required is NIDS sensor software.
- 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.

Passive NIDS

- 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.**



Working of NIDS

1. Traffic Capture: Network packets are captured using port mirroring or taps.

2. Preprocessing: Packets are normalized and reassembled.

3. Detection Engine:

Uses: **Signature-based Detection**: Matches packets with known attack signatures.

Anomaly-based Detection: Detects deviations from normal behavior.

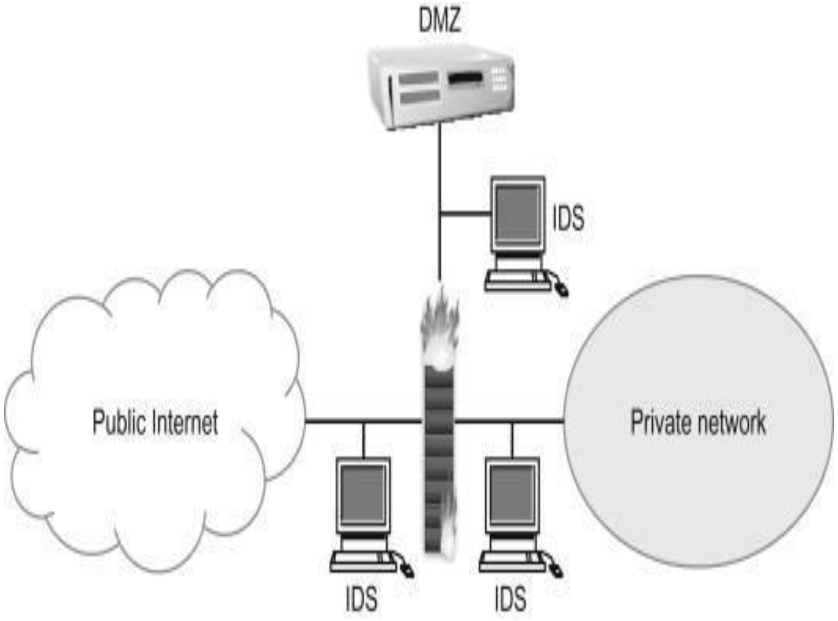
4. Alerting/Logging: Logs and alerts are generated for suspected attacks.

NIDS

- require **promiscuous network access** in order to analyze all traffic, including all uncast traffic.
- **sniffs the internal interface of the firewall in read-only mode** and sends alerts to a NIDS Management server via a different (ie, read/write) network interface.
- **passively inspect traffic traversing the devices** on which they sit.
- can be **hardware or software-based systems**
- NIDS have two network interfaces. One is used for **listening to network conversations** in promiscuous mode and the other is used for **control and reporting.**

Why IDS near DMZ

- Monitors **high-risk public-facing servers**
- **Detects attacks early** before reaching internal network
- Provides visibility into inbound & outbound traffic
- Works with firewall to catch hidden threats
- **Alerts on compromised DMZ hosts attempting lateral movement**



is a special, controlled network segment that adds an extra layer of security between an organization’s internal network and the external, untrusted network (like the internet).

How NIDS analyze threats

- **Packet headers**
- Packet headers contain specific information about the packet being transmitted across your network. **Information in the header can include source and destination IP addresses, ports, protocol types, etc.** The NIDS will analyze this information for suspicious activity or malicious behavior.

How NIDS analyze threats

- **Packets/transmissions**
- **The total packets per second are a common technology** used by a NIDS to monitor for threats on your network. This may be a configuration option that you specify when installing a traffic monitoring system on your network. **IDS can compare normal traffic rates, with those being transmitted at any one time across the network, to detect anything out of the ordinary.**
- **For example, if there is no heavy traffic on the network, but packets are still being transmitted at a high rate of speed, this could indicate suspicious activity.**

How NIDS analyze threats

- **Protocols and applications**
- **Packet protocols**. TCP/IP, UDP/IP, ICMP etc.
- **Anomaly-based protocols**. This is where an IDS has been programmed to detect anomalies in protocols that are otherwise benign when working normally.
- **For example, if you had a specific protocol that was known to have 50% packet loss during normal operation and a packet loss percentage significantly different from the norm is detected-Ex: UDP-based streaming (e.g., IPTV or VoIP over a poor wireless link),** this would trigger an alarm or alert enabling you to investigate the problem further.

How NIDS analyze threats

- **Data flow analysis.** NIDS can analyze data flow throughout the network to determine where a problem may be taking place.
- **For example, if a user suddenly begins transmitting a large amount of data, an IDS will recognize this and alert you to possible security breaches occurring on your network.**

Deployment Locations

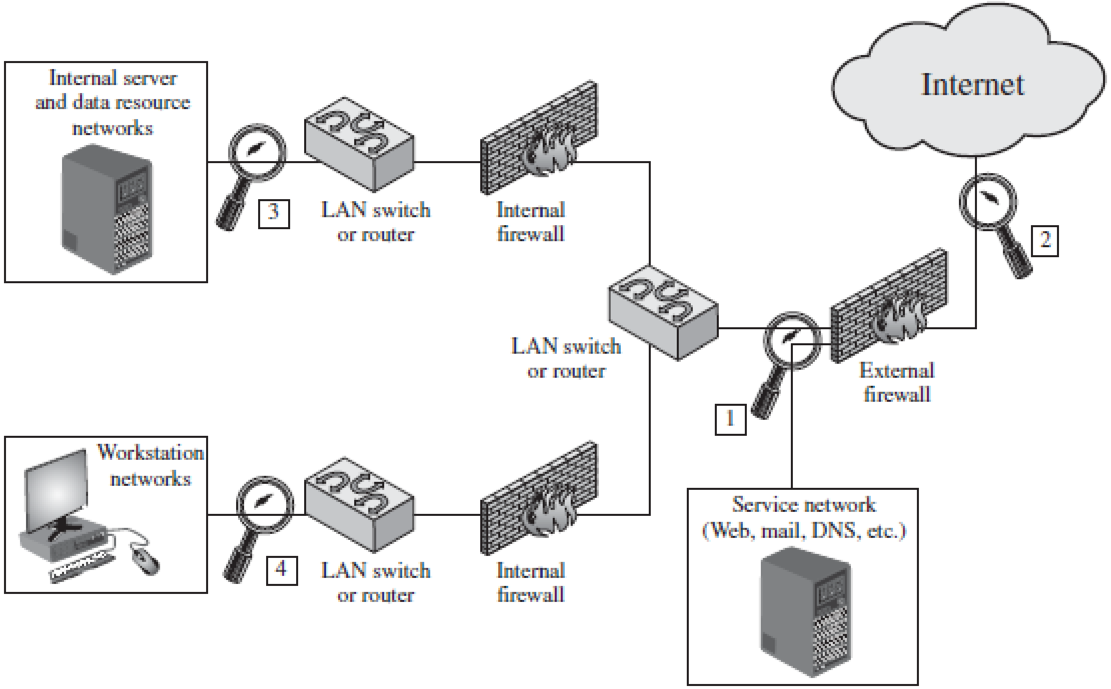


Figure 8.5 **Example of NIDS Sensor Deployment**

Location 1: Behind each external firewall, in the network DMZ

- **Advantages:**
- **Sees attacks, originating from the outside world, that penetrate the network's perimeter defenses.**
- Highlights problems with the network firewall policy or performance
- **Sees attacks that might target the web server or ftp server**, which commonly reside in this DMZ
- Even if the incoming attack is not recognized, the IDS can sometimes recognize the outgoing traffic that results from the compromised server

Location 2: Outside an external firewall

- **Advantages:**
- **Documents number of attacks originating on the Internet that target the network.**
- Documents types of attacks originating on the Internet that target the network

Location 3: On major network backbones

- **Advantages:**
- **Monitors a large amount of a network’s traffic**, thus increasing the possibility of spotting attacks.
- **Detects unauthorized activity by authorized users** within the organization’s security perimeter.

Location 4: On critical subnets

Advantages:

- **Detects attacks targeting critical systems and resources.**
- Allows focusing of limited resources to the network assets considered of greatest value.

Common types of network intrusion detection systems

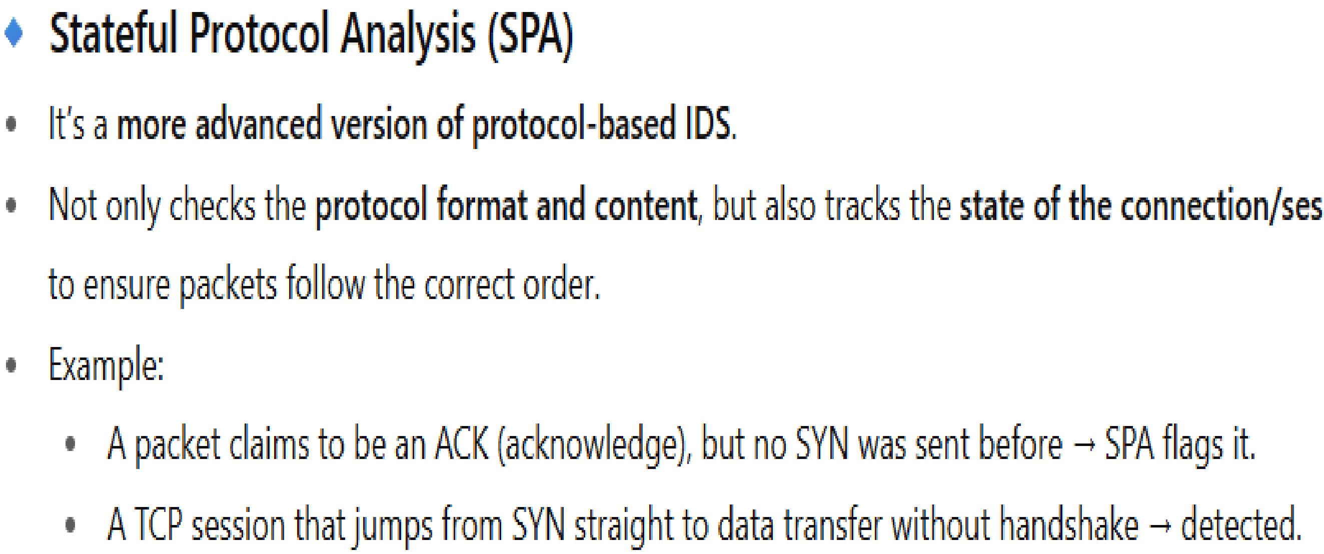
- **Signature-based system**
- This type of NIDS uses signatures from previously analyzed attacks. **It learns which patterns indicate malicious activity so future events with similar characteristics** will be detected immediately. Signature-based systems do not need any knowledge about the normal behavior of users or applications to operate.

Common types of network intrusion detection systems

- **Stateful protocol analysis system**
- This type of NIDS is similar to a signature-based system in that it learns which patterns indicate malicious activity.
- Stateful protocol analysis systems differ because they **do not need to know what specific attacks look like before they are detected.**
- Instead, it can **maintain temporary information about how your network normally operates and will compare new events against the normal traffic rate of existing connections.**

Common types of network intrusion detection systems

- Ex:



♦ Stateful Protocol Analysis (SPA)

- It's a **more advanced version of protocol-based IDS**.
- Not only checks the **protocol format and content**, but also tracks the **state of the connection/session** to ensure packets follow the correct order.
- Example:
  - A packet claims to be an ACK (acknowledge), but no SYN was sent before → SPA flags it.
  - A TCP session that jumps from SYN straight to data transfer without handshake → detected.

Common types of network intrusion detection systems

- **Behavioral-based system**
- This type of NIDS uses behavioral analysis to determine whether any suspicious activity has occurred.
- **If the behavior being analyzed meets certain conditions set by the administrator, an alert will be triggered so appropriate action can be taken in response to malicious activity.**

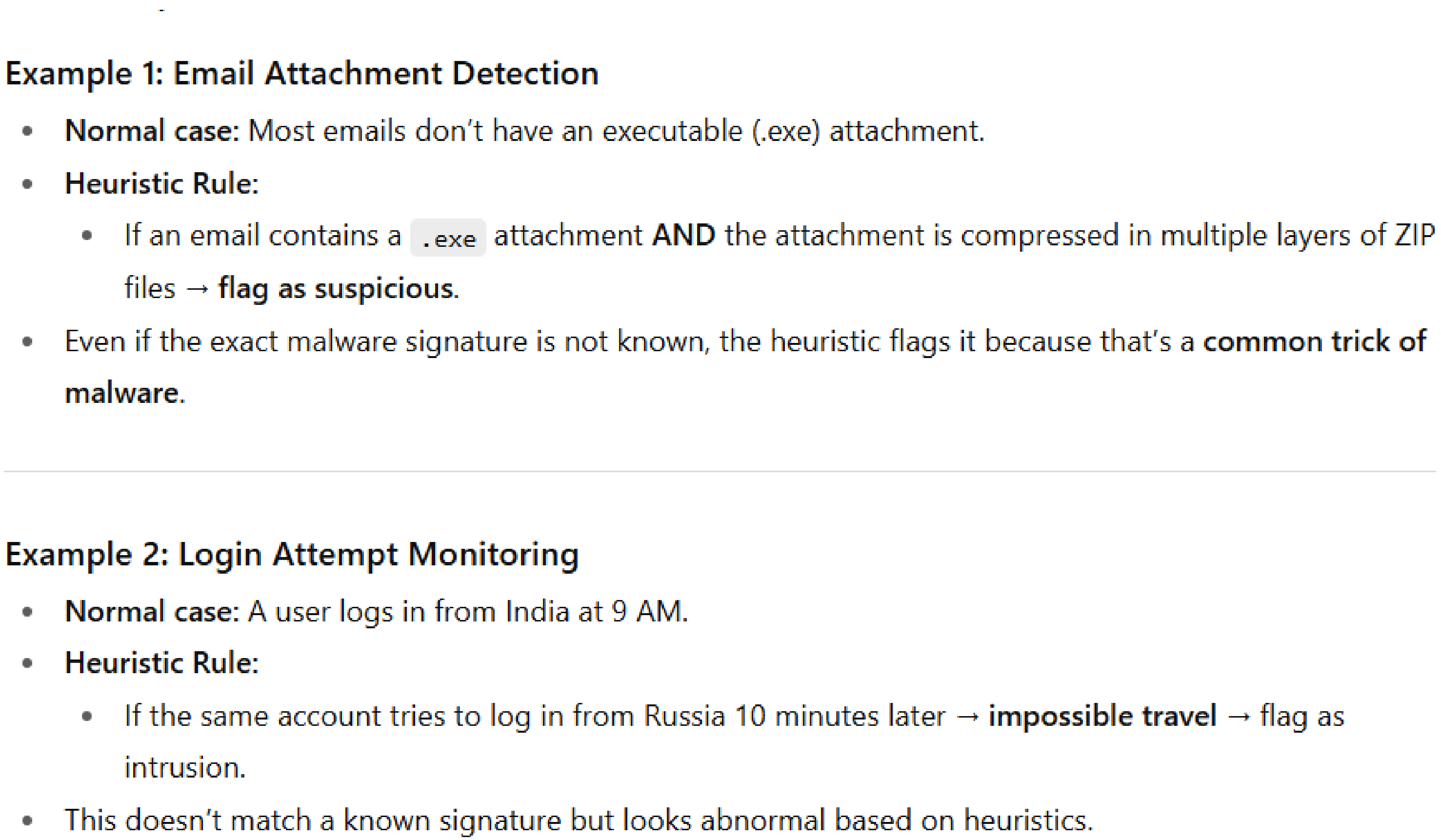
Common types of network intrusion detection systems

- **Anomaly-based system**
- This type of NIDS is similar to the behavior-based system, except that it learns what typical network behavior looks like by analyzing how real connections are established and used over time.
- The administrator may also need to provide information about which events should trigger alerts if anomalies are detected.
- **This type of system is configured to learn what the normal traffic on your network looks like, which can reduce false-positive rates, however, changes in user computer activity or changes made by new software installations could also trigger false alarms.**

Common types of network intrusion detection systems

- **Heuristic-based system**
- This type of NIDS uses heuristics to **look beyond attacks with known signatures** and analyze them against a set of rules to determine whether any suspicious activity has occurred.
- The heuristic-based system is capable of detecting advanced attacks without previously knowing what those attacks look like **by looking for a combination of characteristics that indicate a possible security issue.**

Common types of network intrusion detection systems



Example 1: Email Attachment Detection

- **Normal case:** Most emails don't have an executable (.exe) attachment.
- **Heuristic Rule:**
    - If an email contains a `.exe` attachment **AND** the attachment is compressed in multiple layers of ZIP files → **flag as suspicious**.
- Even if the exact malware signature is not known, the heuristic flags it because that's a **common trick of malware**.

Example 2: Login Attempt Monitoring

- **Normal case:** A user logs in from India at 9 AM.
- **Heuristic Rule:**
    - If the same account tries to log in from Russia 10 minutes later → **impossible travel** → flag as intrusion.
- This doesn't match a known signature but looks abnormal based on heuristics.

Advantages and disadvantages of network intrusion detection system

Advantages

- Detects known and unknown malware.
- Reduces downtime.
- Prevents attacks.
- Detects compromised devices.

Disadvantages

- Requires frequent updating.
- Requires extensive configuration.
- Requires maintenance.

Comparison	NIDS (Network Intrusion Detection System)	Other Technology
NIDS vs. NIPS	Passive system; monitors and compares traffic with known malware signatures.	NIPS is active; analyzes traffic in real-time and blocks suspicious activities, even without full knowledge of threats.
NIDS vs. Firewall	Analyzes data packets for signs of attacks or malicious activities.	Controls traffic based on rules; allows/denies access by IP or computer.
NIDS vs. HIPS (Host-based Intrusion Prevention System)	Monitors suspicious activity across entire network (all computers and servers).	HIPS monitors and blocks suspicious activity on a single host computer.
NIDS vs. Virus Protection	Monitors network traffic to detect suspicious activities before malware spreads.	Detects and removes viruses after they are already downloaded onto a system.
NIDS vs. Anti-Virus Software	Scans all network traffic for suspicious patterns (e.g., port scans, brute-force attacks). Updates automatically.	Protects single hosts; scans files for known malware signatures. Requires manual updates.
NIDS vs. ABIDS (Anomaly-Based IDS)	Uses signatures to detect known malicious activities. Processes only potentially malicious packets.	Uses statistical analysis to detect unusual or anomalous behavior. Better against zero-day attacks but may cause false positives.
NIDS vs. Anomaly-Based IPS	Only detects and alerts suspicious traffic.	Detects anomalies and automatically blocks suspicious activities or shuts down compromised processes. Strong against zero-day attacks but risk of blocking legitimate traffic.

Logging of Alerts

Typical information logged by a NIDS sensor includes the following:

- Timestamp (usually date and time)
- Connection or session ID (typically a consecutive or unique number assigned to each TCP connection or to like groups of packets for connectionless protocols)
- Event or alert type
- Rating (e.g., priority, severity, impact, confidence)
- 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 (e.g., authenticated username)

Host based IDS

Host-based IDSs

- Host-based IDSs add a specialized layer of security software to vulnerable or sensitive systems; **examples include database servers and administrative systems.**
- The primary benefit of a host-based IDS is that it can **detect both external and internal intrusions**, something that is not possible either with network-based IDSs or firewalls.

Host-based IDSs

- **1. Anomaly detection: Involves the collection of data relating to the behavior of legitimate users over a period of time.**
- Then **statistical tests** are applied to observed behavior to determine with a high level of confidence whether that behavior is not legitimate user behavior.

Host-based IDSs

- Two approaches to statistical anomaly detection

a. Threshold detection: This approach involves defining thresholds, independent of user, for the frequency of occurrence of various events.

b. Profile based: A profile of the activity of each user is developed and used to detect changes in the behavior of individual accounts.

Host-based IDSs

2. Signature detection: Involves an attempt to define a set of rules or attack patterns that can be used to decide that a given behavior is that of an intruder.

In practice, a system may employ a combination of both approaches to be effective against a broad range of attacks.

Audit Records Types

1. Native audit records are the **original log entries generated by a system's operating system, database, or application without being altered or reformatted**.

- Virtually **all multiuser operating systems include accounting software that collects information on user activity.**
- The **advantage** of using this information is that no additional collection software is needed.
- The **disadvantage** is that the native audit records may not contain the needed information or may not contain it in a convenient form.

Audit Records Types

- **On Linux / Unix**

System log files → /var/log/ directory

SSH logins:

```
/var/log/auth.log      # Debian/Ubuntu
/var/log/secure        # RedHat/CentOS
```

System Activity:

```
/var/log/syslog        # General logs
/var/log/messages      # System messages
```

Audit Records Types

- **On Windows**

Stored in Windows Event Viewer

Press Windows + R, type eventvwr.msc, press Enter

Navigate to:

Windows Logs → Security # Login, file access events

Windows Logs → System # System-level events

Audit Records Types

- **For Databases**

Oracle:

If auditing is enabled, audit trail is in SYS.AUD$ table or OS log files:

$ORACLE_HOME/rdbms/audit/

- **Web Servers**

Apache:

```
/var/log/apache2/access.log   # Debian/Ubuntu
/var/log/httpd/access_log     # RedHat/CentOS
```

Audit Records Types

2. Detection-specific audit records: A collection facility can be implemented that **generates audit records containing only that information required by the IDS.**

- **Advantage:** made vendor independent and ported to a variety of systems.
- **Disadvantage:** is the extra overhead involved in having, in effect, two accounting packages running on a machine.

Audit Record Elements

- Each audit record contains the following fields:

1. Subject: **Initiators of actions.** A subject is typically a terminal user but might also be a process acting on behalf of users or groups of users.

2. Action: **Operation performed by the subject on or with an object**; for example, login, read, perform I/O, execute.

3. Object: **Receptors of actions.** Examples include files, programs, messages, records, terminals, printers, and user- or program-created structures.

Audit Record

4. Exception-Condition: Denotes which, if any, exception condition is raised on return.

5. Resource-Usage: A list of quantitative elements in which each element gives the amount used of some resource (e.g., number of lines printed or displayed, number of records read or written, processor time, I/O units used, session elapsed time).

6. Time-Stamp: Unique time-and-date stamp identifying when the action took place.

Anomaly Detection in Host based IDS

- **Threshold detection involve counting the number of occurrences of a specific event type over an interval of time.** If the count surpasses what is considered a reasonable number that one might expect to occur, then intrusion is assumed. Threshold analysis, by itself, is a crude and ineffective detector of even moderately sophisticated attacks.

Anomaly Detection in Host based IDS

- **Profile-based anomaly detection** focuses on characterizing the past behavior of individual users or related groups of users and then detecting significant deviations. ***A profile may consist of a set of parameters, so that deviation on just a single parameter may not be sufficient in itself to signal an alert.***

Analysis of audit records for Profile based Anomaly Detection

- First, the designer **must decide on a number of quantitative metrics that can be used to measure user behavior.** An analysis of audit records over a period of time can be used to determine the activity profile of the average user. Thus, the audit records serve to define typical behavior.
- Second, **current audit records are the input used to detect intrusion. That is, the intrusion detection model analyzes incoming audit records to determine deviation from average behavior.**

Metrics for Profile based Anomaly Detection

- **Counter: A nonnegative integer** that may be incremented but not decremented until it is reset by management action. Typically, a count of certain event types is kept over a particular period of time.

Examples include the number of logins by a single user during an hour, the number of times a given command is executed during a single user session, and the number of password failures during a minute.

- **Gauge:** A nonnegative integer that may be incremented or decremented. **Typically, a gauge is used to measure the current value of some entity**.

Examples include the number of logical connections assigned to a user application and the number of outgoing messages queued for a user process.

- **Resource utilization: Quantity of resources consumed** during a specified period.

Examples include the number of pages printed during a user session and total time consumed by a program execution.

Statistical tests over Profile based metrics collected

- **Mean and standard deviation:** gives a reflection of the average behavior and its variability. The use of mean and standard deviation is applicable to a wide variety of **counters, timers, and resource measures.**
- **Multivariate:** based on correlations between two or more variables.Intruder behavior may be characterized with greater confidence by considering such correlations **(e.g., processor time and resource usage, or login frequency and session elapsed time).**
- **Markov process:**

A **Markov process** is a way to model **sequences of events**, where:

The **next step** depends **only on the current step** (not the full history).

You assign **probabilities** to moving from one state to another.

Where the Markov process comes in

A **Markov process** is a way to model **sequences of events**, where:

- The **next step** depends **only on the current step** (not the full history).
- You assign **probabilities** to moving from one state to another.

Why it's useful here

In profile-based anomaly detection, you don't just care about **which actions happen** — you also care about the **order in which they happen.**

Example:

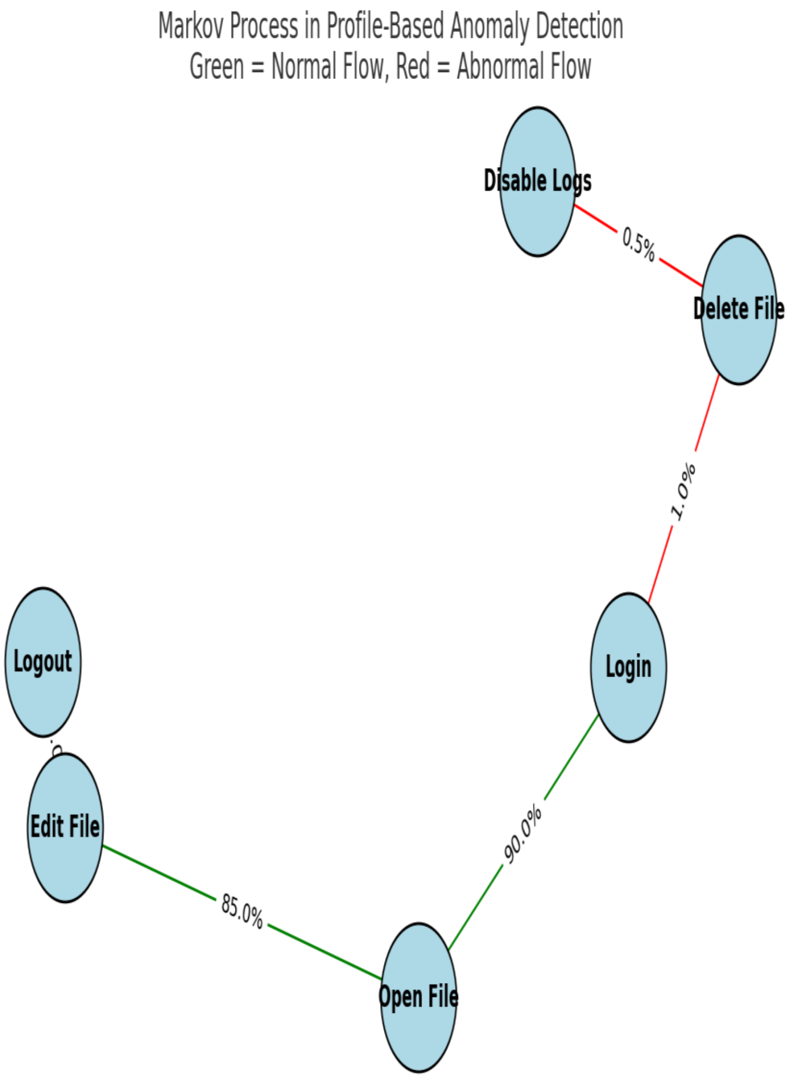
- A normal user might have this command sequence:

  `login → open file → edit file → logout`
- In a Markov model, each step has a **probability**:
  - From `login` → `open file`: 90% chance
  - From `login` → `delete file`: 1% chance

If one day you see:

`login → delete file → disable logs → logout`

The probability of this sequence under the normal profile is **very low**, so it's flagged as **anomaly**.



Markov Process in Profile-Based Anomaly Detection
Green = Normal Flow, Red = Abnormal Flow

Statistical tests over Profile based metrics collected

- **Time series:** model focuses on time intervals, looking for sequences of events that happen too rapidly or too slowly.
- **Operational:** is based on a judgment of **what is considered abnormal, rather than an automated analysis of past audit records.**

Typically, fixed limits are defined and intrusion is suspected for an observation that is outside the limits. This type of approach works best where intruder behavior can be deduced from certain types of activities.

For example, a large number of login attempts over a short period suggests an attempted intrusion.

Measure	Model	Type of Intrusion Detected
Login and Session Activity		
Login frequency by day and time	Mean and standard deviation	Intruders may be likely to log in during off hours.
Frequency of login at different locations	Mean and standard deviation	Intruders may log in from a location that a particular user rarely or never uses.
Time since last login	Operational	Break-in on a “dead” account.
Elapsed time per session	Mean and standard deviation	Significant deviations might indicate masquerader.
Quantity of output to location	Mean and standard deviation	Excessive amounts of data transmitted to remote locations could signify leakage of sensitive data.
Session resource utilization	Mean and standard deviation	Unusual processor or I/O levels could signal an intruder.
Password failures at login	Operational	Attempted break-in by password guessing.
Failures to login from specified terminals	Operational	Attempted break-in.
Command or Program Execution Activity		
Execution frequency	Mean and standard deviation	May detect intruders, who are likely to use different commands, or a successful penetration by a legitimate user, who has gained access to privileged commands.
Program resource utilization	Mean and standard deviation	An abnormal value might suggest injection of a virus or Trojan horse, which performs side effects that increase I/O or processor utilization.
Execution denials	Operational model	May detect penetration attempt by individual user who seeks higher privileges.
Fil Access Activity		
Read, write, create, delete frequency	Mean and standard deviation	Abnormalities for read and write access for individual users may signify masquerading or browsing.
Records read, written	Mean and standard deviation	Abnormality could signify an attempt to obtain sensitive data by inference and aggregation.
Failure count for read, write, create, delete	Operational	May detect users who persistently attempt to access unauthorized files.

Signature Detection in Host based IDS

- Signature techniques detect intrusion by observing events in the system and **applying a set of rules that lead to a decision regarding whether a given pattern of activity is or is not suspicious.**
- In very general terms, we can characterize all approaches as focusing on either anomaly detection or penetration identification, although there is **some overlap in these approaches.**

Rule-based anomaly detection in Host based IDS

- With the rule-based approach, **historical audit records are analyzed to identify usage patterns and to generate automatically rules that describe those patterns.**
- Rules may represent past behavior patterns of users, programs, privileges, time slots, terminals, and so on. **Current behavior is then observed, and each transaction is matched against the set of rules to determine if it conforms to any historically observed pattern of behavior.**

Rule-based penetration identification in Host based IDS

- **Rule-based penetration identification** takes a very different approach to intrusion detection. The key feature of such systems is the **use of rules for identifying known penetrations or penetrations that would exploit known weaknesses.**
- **Used to analyze attack tools and scripts** collected on the Internet.
- Can be supplemented with rules generated by knowledgeable security personnel.

Rule-based penetration identification in Host based IDS

- Example heuristics based rules

1. Users should not read files in other users’ personal directories.

2. Users must not write other users’ files.

3. Users who log in after hours often access the same files they used earlier.

4. Users do not generally open disk devices directly but rely on higher-level operating system utilities.

5. Users should not be logged in more than once to the same system.

6. Users do not make copies of system programs

Rule-based penetration identification in Host based IDS

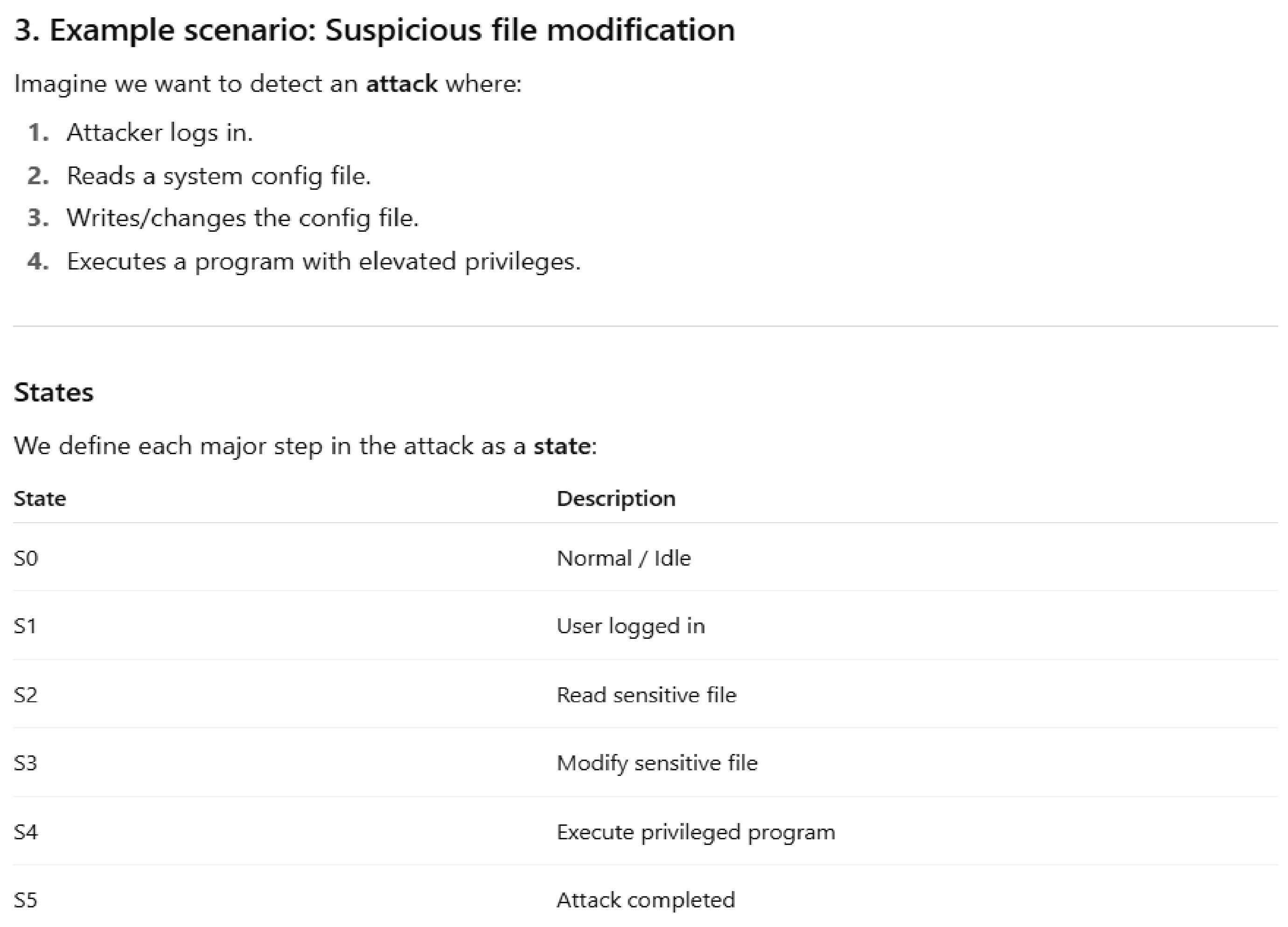
- Audit records are examined as they are generated, and they are matched against the rule base. **If a match is found, then the user's *suspicion rating* is increased.** If enough rules are matched, then the rating will pass a threshold that results in the reporting of an anomaly.

UNIX State Transition Analysis Tool.

- It’s a specialized implementation of the generic **STAT (State Transition Analysis Tool)** technique, adapted specifically for **Solaris’s Basic Security Module (BSM)** audit data
- Instead of tracking every small **audit record** (hundreds of different log events), we group them into a few **general actions** (like *login*, *read file*, *write file*, *execute program*).
- Then, we **model attacks as a sequence of these actions**.
- We use a **state transition diagram** to see if the observed actions match a known attack path.

Table 8.3 USTAT Actions versus SunOS Event Types

USTAT Action	SunOS Event Type
Read	open_r, open_rc, open_rtc, open_rwc, open rwtc, open_rt, open_rw, open_rwt
Write	truncate, ftruncate, creat, open_rtc, open_rwc, open_rwtc, open_rt, open_rw, open_rwt, open_w, open_wt, open_wc, open_wct
Create	mkdir, creat, open_rc, open_rtc, open_rwc, open_rwtc, open_wc, open_wtc, mknod
Delete	rmdir, unlink
Execute	exec, execve
Exit	exit
Modify_Owner	chown, fchown
Modify_Perm	chmod, fchmod
Rename	rename
Hardlink	link



3. Example scenario: Suspicious file modification

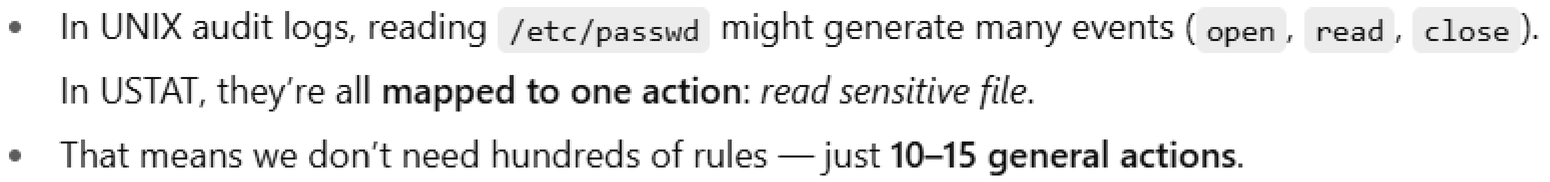
Imagine we want to detect an **attack** where:

1. Attacker logs in.
2. Reads a system config file.
3. Writes/changes the config file.
4. Executes a program with elevated privileges.

States

We define each major step in the attack as a **state**:

State	Description
S0	Normal / Idle
S1	User logged in
S2	Read sensitive file
S3	Modify sensitive file
S4	Execute privileged program
S5	Attack completed



- In UNIX audit logs, reading `/etc/passwd` might generate many events (`open`, `read`, `close`). In USTAT, they're all **mapped to one action**: *read sensitive file*.
- That means we don't need hundreds of rules — just **10–15 general actions**.

base-rate fallacy

- IDS **should detect a substantial percentage of intrusions** while keeping the false alarm rate at an acceptable level.
- If only a modest percentage of actual intrusions are detected, the system provides a false sense of security.
- On the other hand, if the system **frequently triggers an alert when there is no intrusion** (a false alarm), then either **system managers will begin to ignore the alarms,** or much time will be wasted analyzing the false alarms.

base-rate fallacy

- because of the nature of the probabilities involved, it is **very difficult to meet the standard of high rate of detections** ***with a low rate of false alarms.***
- In general, **if the actual numbers of intrusions is low compared to the number of legitimate uses of a system, then the false alarm rate will be high** unless the test is extremely discriminating→*base-rate fallacy*

Protocol based IDS

PROTOCOL BASED IDS

- Type of IDS that ***monitors and analyzes the application-layer protocol traffic*** to ensure it follows predefined standards and behavior.
- PIDS focuses on specific protocols like ***HTTP, FTP, SMTP, or DNS*** and verifies whether the communication aligns with the protocol specifications
- filter or **gatekeeper that sits between the external client and the application server**, analyzing protocol-specific interactions in real time.

HTTP Protocol-Based IDS

- Let's consider a **web server** running an e-commerce website. A user interacts with this website using the **HTTP protocol** (via browsers or mobile apps). The attacker might try to inject a **malicious SQL query** in a form field to extract sensitive information from the backend database.

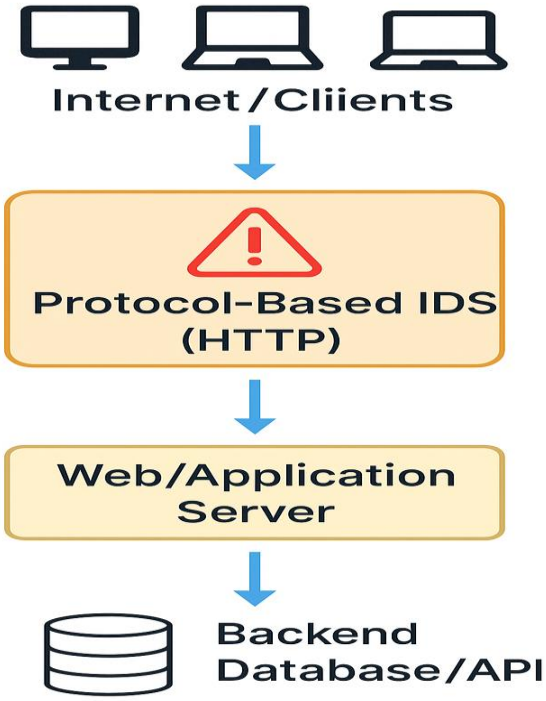
HTTP Protocol-Based IDS

- http
- GET /search.php?query=shoes'+OR+1=1-- HTTP/1.1
- Host: example.com
- This request contains a **SQL Injection attempt**: shoes' OR 1=1-- is an SQL condition that always evaluates to true and could potentially expose the entire product database or user information.

How **PIDS Detects This**

- A **Protocol-Based IDS for HTTP** inspects the query string.
- It **parses the HTTP GET request**, identifies abnormal use of characters (like ', --, or OR 1=1), and flags it as **malicious behavior** based on:
  - **Known signature patterns**
  - **Deviation from normal query behavior**
- An alert is triggered, and the request is blocked **before it reaches the web server.**

Architecture of Protocol based IDS



Components in the Architecture

- **Clients/Users:** Initiate communication using application protocols.
- **PIDS:** Acts as a protocol-aware security layer, inspecting requests/responses.
- **Web/Application Server:** Executes requests if verified safe by PIDS.
- **Backend Systems:** Handle final data processing or business logic.

Key Functions of PIDS

- **Protocol Parsing:** Understands and analyzes the structure of HTTP, FTP, DNS, etc.
- **Signature Matching:** Compares request patterns against known attack signatures.
- **Anomaly Detection:** Flags deviations from normal usage, even if unknown attacks.
- **Real-Time Protection:** Acts before the request reaches the application.
- **Logging & Alerting:** Records suspicious activity for review and response.

Hybrid IDS

HYBRID IDS

- **combines two or more different types of intrusion detection approaches**—primarily signature-based and anomaly-based detection systems, and may also integrate host-based and network-based detection components.

HYBRID IDS

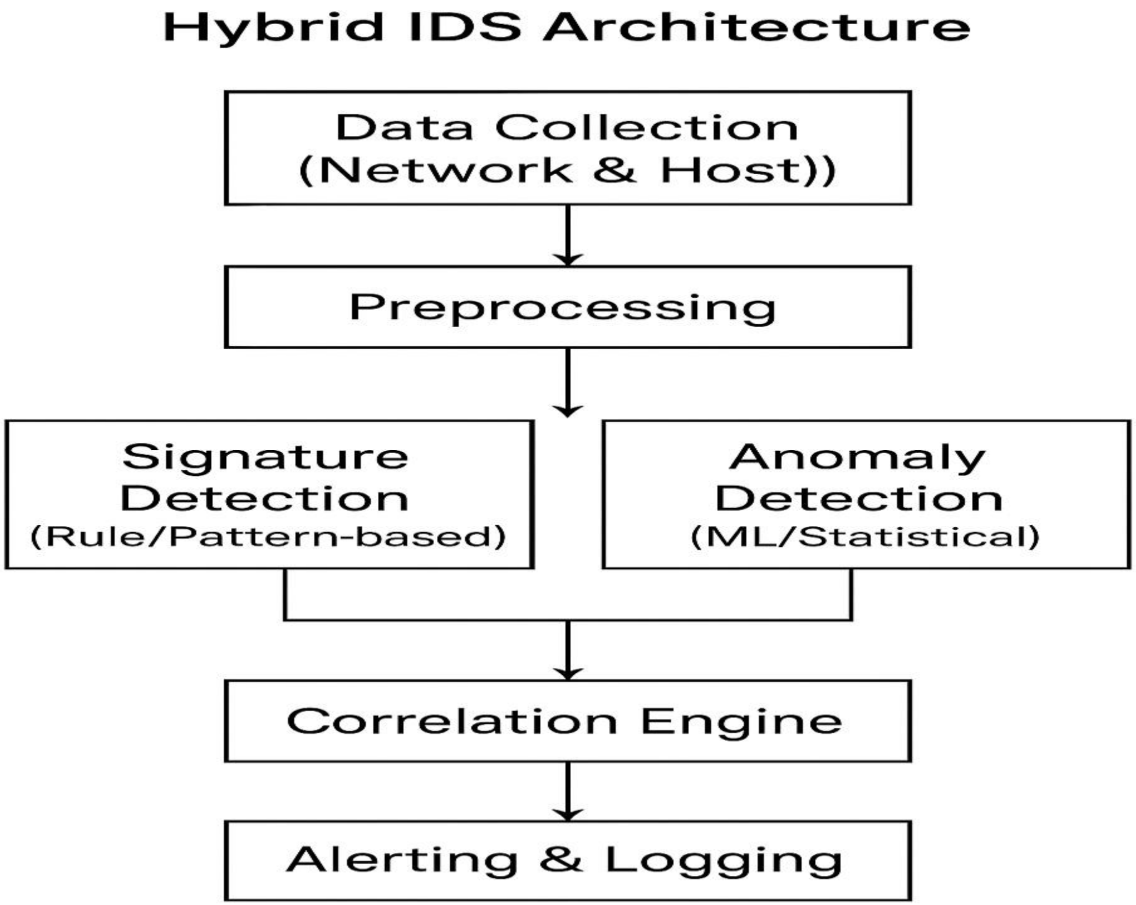
IDS Type	Strengths	Weaknesses
Signature-based	Accurate for known attacks	Can't detect new/unknown threats
Anomaly-based	Can detect unknown attacks	High false positive rate
Host-based IDS (HIDS)	Monitors system-level activities	Limited to specific host
Network-based IDS (NIDS)	Monitors entire network	May miss host-specific anomalies

HYBRID IDS

A **Hybrid IDS** combines these to:

- Improve detection accuracy
- Reduce false positives
- Provide both breadth (network) and depth (host-level) monitoring

Hybrid IDS


Hybrid IDS Architecture
Data Collection (Network & Host))
Preprocessing
Signature Detection (Rule/Pattern-based)
Anomaly Detection (ML/Statistical)
Correlation Engine
Alerting & Logging

Hybrid IDS Architecture

🔷 **Data Collection Module**

- Captures logs and traffic data from hosts and network.
- **Examples: System calls, packet captures, logs**

🔷 **Preprocessing Module**

- Filters and normalizes data
- Extracts relevant features

Hybrid IDS Architecture

🔷 **Detection Engine**

- Uses **Signature Matching** **(for known attacks)**
- Uses **Anomaly Detection** (**for unknown attacks)**
- Often integrates **machine learning models** or **statistical methods**

🔷 **Correlation Engine**

- Combines outputs from HIDS and NIDS
- Correlates alerts across multiple sources

Hybrid IDS Architecture

🔷 Alerting and Reporting

- Generates alerts for administrators
- Provides dashboards, logs, or real-time notifications

Hybrid IDS-Example

- **Signature-based engine** detects a known SQL Injection attack.
- **Anomaly-based engine** detects an unusual login time and large data exfiltration from a user account (a possible insider threat).
- **Correlation engine** links both events and flags it as a **high-severity incident**.
- This synergy between components helps detect **complex attacks** that may otherwise go unnoticed.

Advantages

- Detects **both known and unknown** threats
- Reduces **false positives** by validating anomalies with signatures
- Combines **host and network level** insights
- Supports **adaptive learning** using machine learning algorithms

Intrusion Prevention Systems

INTRUSION PREVENTION SYSTEMS

- An Intrusion Prevention System (IPS) is a network security technology that **monitors network traffic for malicious activity and automatically blocks or prevents it**. It acts as a proactive defense, sitting inline with network traffic to detect and respond to threats in real-time.

Key Features and Functions

Real-time Monitoring:

- IPS continuously monitors network traffic for suspicious activity, looking for patterns and signatures that match known attacks.

Malicious Activity Blocking:

- When a threat is detected, the IPS can automatically block the malicious traffic, preventing it from entering the network.

Alerting and Logging:

- IPS can also generate alerts to notify security teams of detected threats and log relevant information for analysis and incident response.

Customizable Rules:

- IPS systems can be configured with rules and policies to define what constitutes malicious activity and how the system should respond.

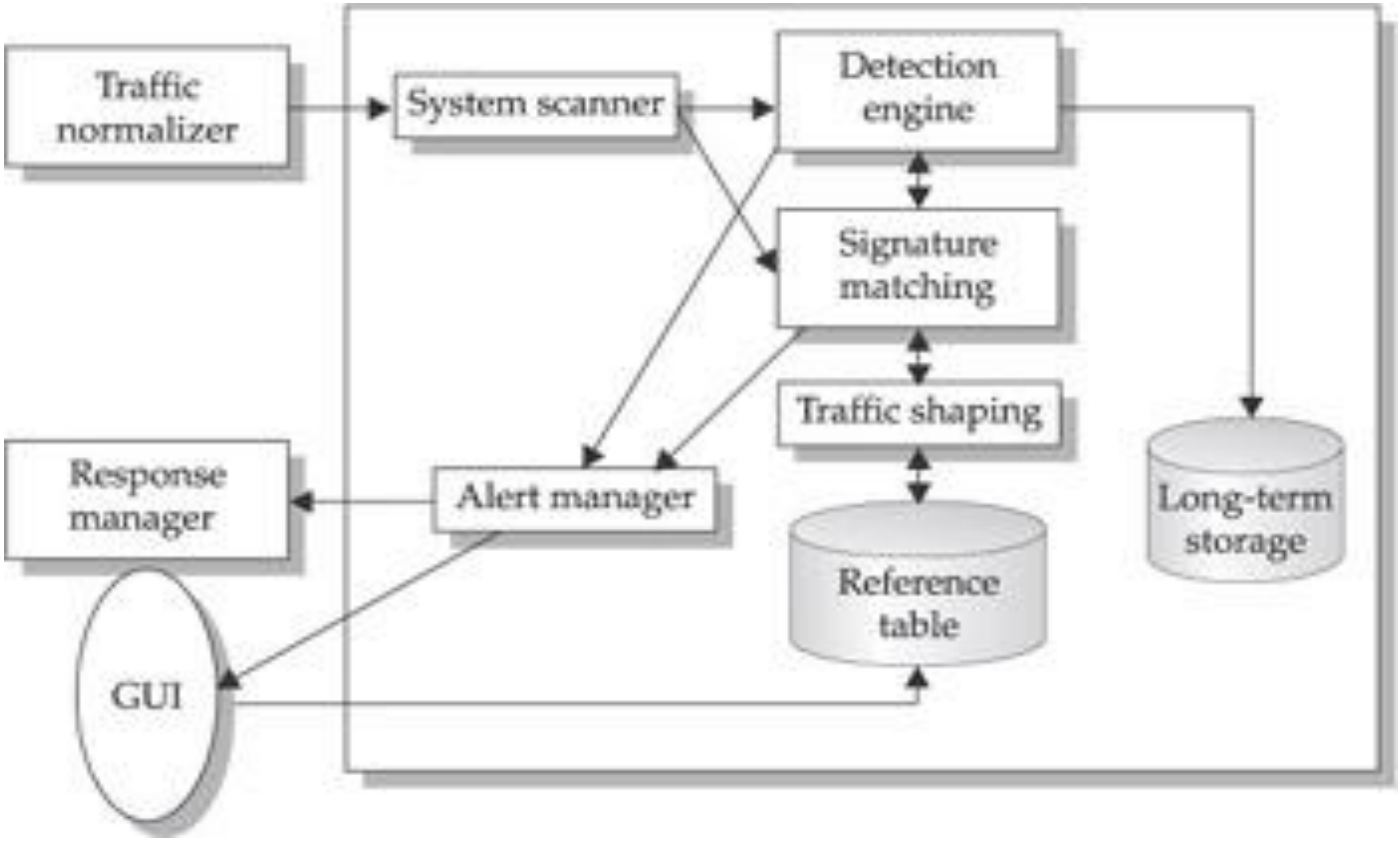
Integration with other security tools:

- IPS can be integrated with other security solutions like firewalls and antivirus software to create a comprehensive security posture.

Common Actions Taken by IPS

- **Blocking malicious traffic:** Dropping packets, resetting connections, or blocking traffic from specific IP addresses.
- **Alerting security personnel:** Sending notifications about detected threats.
- **Terminating dangerous connections**: Stopping malicious connections that are actively being used.
- **Modifying security policies:** Adjusting firewall rules or other security settings to prevent future attacks.

Standard IPS system



Standard IPS system

- **The traffic normalizer** will interpret the network traffic and **do packet analysis and packet reassembly**, as well as performing basic blocking functions.
- The traffic is then fed into the detection engine and the **service scanner**. The **service** scanner builds a reference table that classifies the information and helps the **traffic shaper** manage the flow of the information.
- The **detection engine** does pattern matching against the reference table, and the appropriate response is determined.

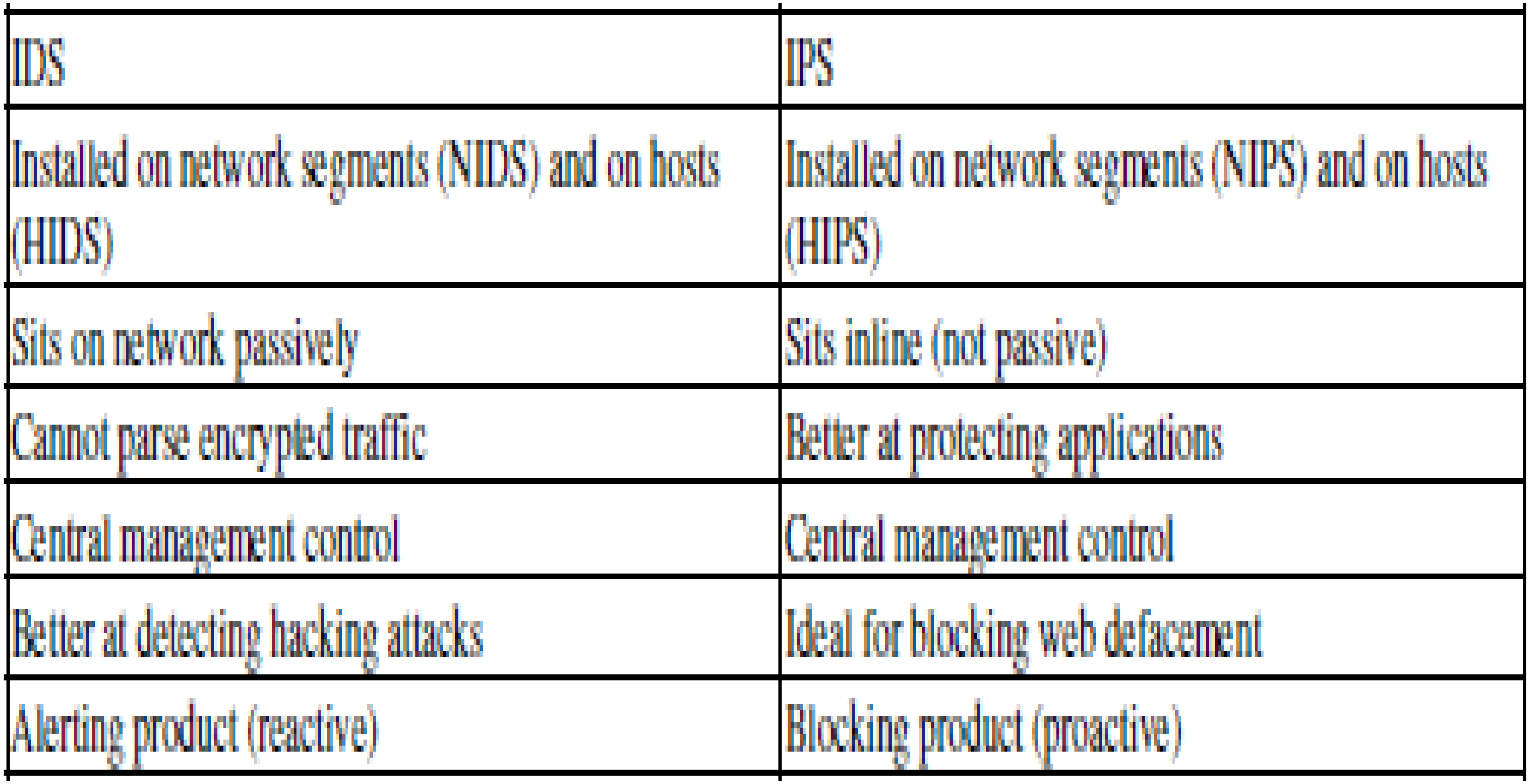
Service Scanner

- Detects active services running on a host (e.g., HTTP, FTP, SSH).
- Identifies service versions to check for vulnerabilities.
- Maps open ports to corresponding services.
- Helps in vulnerability assessment by spotting outdated or misconfigured services.
- Can detect unauthorized services running on the network.
- Used in both security auditing and attack reconnaissance.
- Examples: Nmap service scan, Nessus service detection.

Traffic Shaper

- **Controls the rate** of network traffic flow.
- **Prioritizes** certain types of traffic (e.g., VoIP over file downloads).
- **Ensures fair bandwidth** distribution among users.
- **Prevents network congestion** by limiting data rates.
- Can **delay or drop packets** to enforce limits.
- Often used for Quality of Service (QoS).Helps maintain consistent performance for critical applications.

IDS vs IPS



IDS	IPS
Installed on network segments (NIDS) and on hosts (HIDS)	Installed on network segments (NIPS) and on hosts (HIPS)
Sits on network passively	Sits inline (not passive)
Cannot parse encrypted traffic	Better at protecting applications
Central management control	Central management control
Better at detecting hacking attacks	Ideal for blocking web defacement
Alerting product (reactive)	Blocking product (proactive)

IDS vs IPS vs Firewall

Feature	IDS (Intrusion Detection System)	IPS (Intrusion Prevention System)	Firewall
Primary Role	Monitors and detects suspicious activity	Detects and actively blocks suspicious activity	Controls access based on predefined rules
Placement	Out-of-band (passive monitoring)	Inline (between source and destination)	Inline (perimeter or host-based)
Action	Alerts administrators	Blocks, drops, or modifies malicious traffic	Allows or denies traffic based on IP, port, protocol
Traffic Analysis	Deep packet inspection for threats	Deep packet inspection + prevention	Packet header inspection (basic), some can do deep inspection
Response Time	After attack is detected	Before malicious traffic reaches target	Before traffic enters network segment
Prevention Capability	No (detection only)	Yes (prevention + detection)	Yes (based on rules, not deep behavioral analysis)
Focus	Identifying intrusions	Blocking intrusions in real-time	Enforcing access control
Examples	Snort (IDS mode), Zeek	Snort (IPS mode), Suricata	pfSense, Cisco ASA, FortiGate

Platform	Type	IDS Function	IPS Function	Open Source / Commercial	Notable Features
Snort (Cisco)	Software	Detects threats via signature-based rules	Inline mode to block malicious packets	Open-source (with commercial support)	Widely used, large rule set, flexible deployment
Suricata (OISF)	Software	Signature + protocol + anomaly detection	Inline blocking using NFQUEUE or AF_PACKET	Open-source	Multi-threaded, supports high traffic loads, NSM features
Zeek + NFQUEUE	Software combo	Scriptable IDS for custom analysis	NFQUEUE enables packet drop/block	Open-source	Highly customizable, good for research and custom policies
Cisco Firepower Threat Defense (FTD)	Hardware / Virtual Appliance	Advanced signature & behavioral detection	Real-time prevention with Snort engine	Commercial	NGFW + IPS, integrates with Cisco ecosystem
Palo Alto Networks NGFW	Hardware / Virtual Appliance	Threat detection with App-ID, Content-ID	Inline threat prevention	Commercial	App-aware policies, integrated threat intelligence
Fortinet FortiGate	Hardware / Virtual Appliance	Built-in IDS engine	Inline IPS blocking	Commercial	All-in-one firewall, VPN, antivirus, and IDS/IPS
Security Onion	Linux distro	Suricata/Snort, Zeek for detection	Inline blocking via Suricata IPS mode	Open-source	Bundled with ELK, threat hunting tools, and NSM features

Host based IPS

Host-Based IPS

- **Modification of system resources:** Rootkits, Trojan horses, and backdoors operate by **changing system resources, such as libraries, directories, registry settings, and user accounts.**
- **Privilege-escalation exploits:** These attacks attempt to give ordinary users root access.

Host-Based IPS

- **Buffer-overflow exploits**: A buffer overflow occurs when a program writes more data to a fixed-length block of memory (a buffer) than it can hold. This extra data can overwrite adjacent memory, potentially allowing an attacker to:
- **Execute arbitrary code,**
- **Crash the application,**
- **Escalate privileges.**

This is commonly exploited when input is not properly validated.

- **Access to e-mail contact list:** Many worms spread by mailing a copy of themselves to addresses in the local system’s e-mail address book.

Host-Based IPS

- **Directory traversal:** A directory traversal vulnerability in a Web server allows the hacker to **access files outside the range** of what a server application user would normally need to access.

HIPS typically offers desktop protection

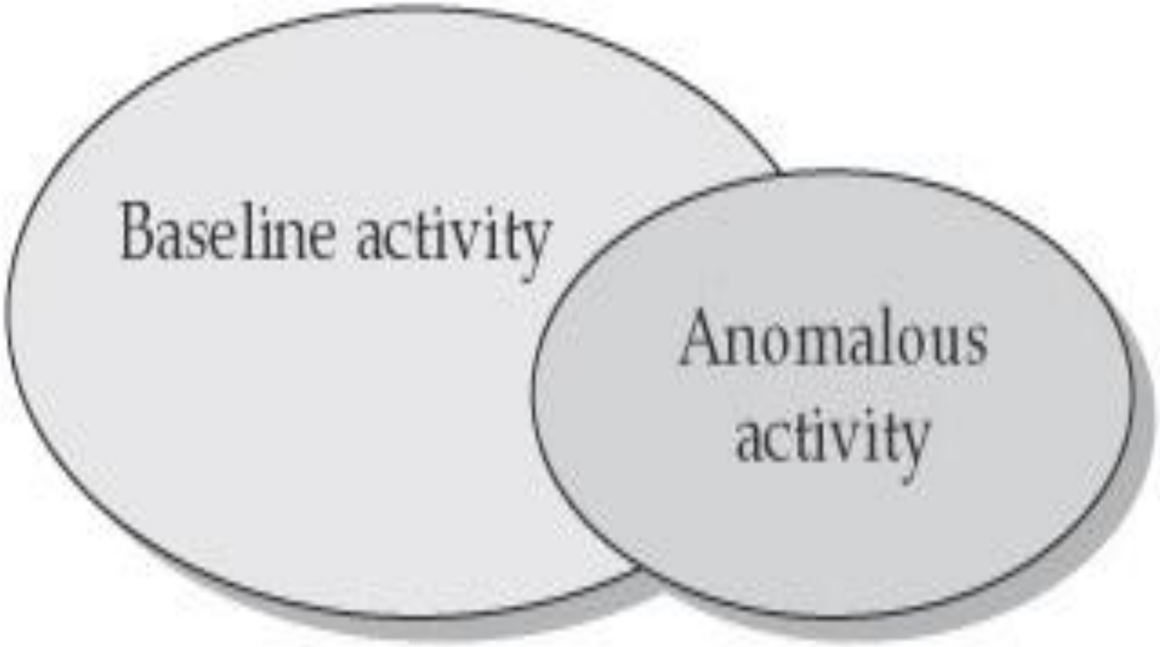
- **System calls:** The kernel controls access to system resources such as memory, I/O devices, and processor. To use these resources, user applications invoke system calls to the kernel. **Any exploit code will execute at least one system call**. ***The HIPS can be configured to examine each system call for malicious characteristics.***
- **File system access:** The HIPS can ensure that file access system calls are not malicious and meet established policy.

HIPS typically offers desktop protection

- **System registry settings:** The registry maintains **persistent configuration information about programs** and is often maliciously modified to extend the life of an exploit. The HIPS can ensure that the **system registry maintains its integrity**.
- **Host input/output:** I/O communications, whether **local or network based can propagate exploit code and malware.** The HIPS can examine and enforce proper client interaction with the network and its interaction with other devices.

Intrusion Analysis

Anatomy of Intrusion Analysis



The relationship between baseline and anomalous network activity

4 Steps in Intrusion-Analysis

1. **Preprocessing:**

- the data are organized in some fashion for classification;
- will help determine the format the data are put into, which is usually some canonical format or could be a structured database.
- if rule-based detection is being used, the classification will involve rules and pattern descriptors

4 Steps in Intrusion-Analysis

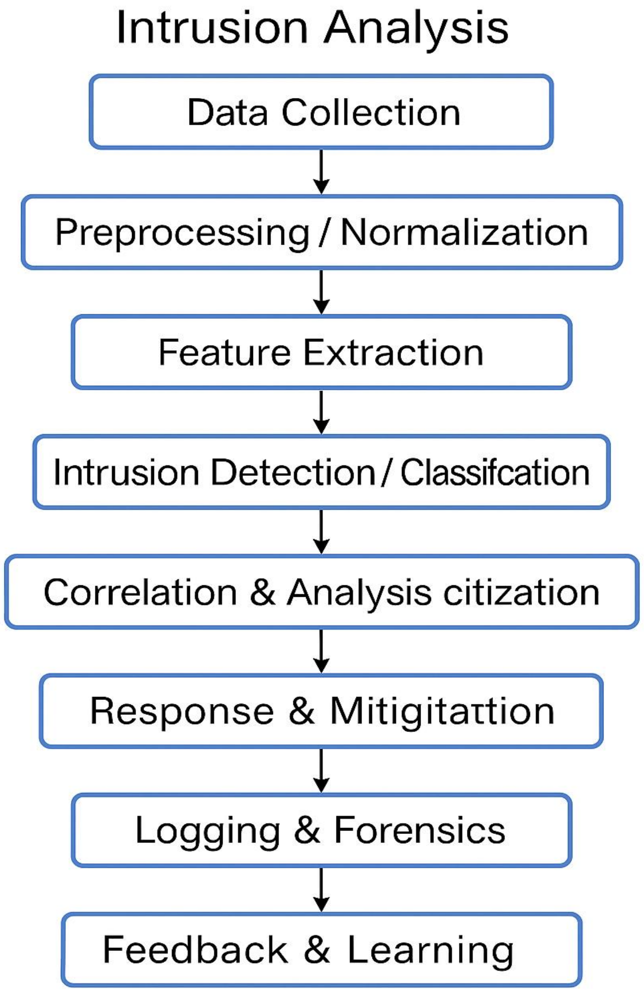
- **2. Analysis:** data record is compared to the knowledge base, and the data record will either be **logged as an intrusion event or it will be dropped**
- **3. Response:** is one of the differentiating factors between IDS and IPS. With IDS, you typically have limited prevention abilities—you are getting the information passively after the fact, so you will have an alert after the fact

4 Steps in Intrusion-Analysis

4. Refinement:

- The fine-tuning of the IDS or IPS system can be done, based on previous usage and detected intrusions.
- This gives the security professional a chance to **reduce false-positive levels** and to have a more accurate security tool.

A MODEL FOR INTRUSION ANALYSIS



Intrusion Analysis

Data Collection

Preprocessing / Normalization

Feature Extraction

Intrusion Detection / Classifcation

Correlation & Analysis citization

Response & Mitigitaτtion

Logging & Forensics

Feedback & Learning

A MODEL FOR INTRUSION ANALYSIS

1. Data Collection

- Gather data from network traffic, system/app logs, user activity, protocol communications.
- Use tools: packet sniffers, system monitors, log aggregators.

2. Preprocessing / Normalization

- Clean and standardize raw data.
- Remove duplicates, handle missing values, convert timestamps, filter irrelevant fields.
- Ensure consistent format for analysis.

3. Feature Extraction

- Identify key indicators: IPs, protocols, request rates, command sequences, auth failures.
- High-quality features boost detection accuracy.

A MODEL FOR INTRUSION ANALYSIS

4. Intrusion Detection / Classification

- **Signature-based:** Match with known patterns.
- **Anomaly-based:** Detect deviations from normal behavior.
- **ML-based:** Use algorithms (Decision Tree, Random Forest, SVM) for prediction.

5. Correlation & Analysis

- Link related events to detect multi-step/coordinated attacks.
- Use threat intelligence for context and enrichment.

6. Response & Mitigation

- **Passive:** Alerts via logs, email, dashboards.
- **Active:** Block IPs, disconnect sessions, isolate devices.
- Map actions to predefined security policies.

A MODEL FOR INTRUSION ANALYSIS

7. Logging & Forensics

- Record all events, actions, and results.
- Aid in forensics, legal evidence, audits, and incident reviews.

8. Feedback & Learning

- Review false positives/negatives.
- Update rules, retrain ML models, refine thresholds.
- Improve adaptability to new threats.

IDS Responses

RESPONSE OPTIONS FOR IDS

Active IDS Responses

Automated actions taken upon detecting intrusions. Categories:

1. Collect Additional Information

- Increase monitoring sensitivity (more logs, full packet capture).
- Helps confirm attacks and gather evidence.
- Supports investigation, apprehension, and legal actions.

2. Change the Environment

- Halt ongoing attacks and block further access.
- Actions:
  - Inject TCP reset packets to terminate connections.
  - Block attacker's IP/site via routers/firewalls.
  - Block targeted ports, protocols, or services.
  - In extreme cases, sever specific network interfaces.

Active IDS Responses

3. Take Action Against the Intruder *(High risk)*

- Includes counterattacks or gathering attacker host info.
- Risks:
  - Legal issues and civil liability.
  - False addresses may harm innocent parties.
  - May escalate attacks.
- Only consider with **human oversight** and **legal advice**.

Passive IDS Responses

Provide information to users; humans decide next actions.

1. Alarms & Notifications

- Onscreen alerts/popups with attack details (source/target IP, tool used, outcome).
- Remote notifications: SMS, pagers, (email discouraged — attackers may monitor).

2. SNMP Traps & Plug-ins

- Send alerts to network management systems via SNMP.
- Benefits:
  - Centralized monitoring.
  - Shift active response load to other systems.
  - Use common comms channels for alerts.

Passive IDS Responses

3. Reporting & Archiving

- Generate routine reports (weekly/monthly).
- Export logs/statistics for databases/reporting tools (e.g., Crystal Reports).

4. Failsafe Considerations

- Protect IDS from detection/circumvention.
- Ensure silent, reliable monitoring.
- Use encryption/authentication for IDS communications.

Type of Response	Nature	Description
Alert	Passive	Sends notification to administrator
Log Event	Passive	Stores the data in logs for later investigation
Connection Termination	Active	Cuts the TCP/UDP connection with the attacker
IP Address Blocking	Active	Prevents further access from attacker’s IP
Session Isolation	Active	Disconnects only the session involved in intrusion
Rate Limiting	Active	Reduces bandwidth for malicious traffic source
Quarantine	Active	Moves the infected host to a separate VLAN or subnet
Honeypot Redirection	Active	Redirects intruder to a trap system

Mapping IDS Responses to Policy

Mapping IDS Responses to Policy

- **Mapping responses to security policies** ensures that the IDS actions are aligned with organizational risk posture and compliance standards.

What is a Security Policy?

- A formal set of rules that governs how systems respond to threats.
- **Defines acceptable use, incident handling, escalation procedures, and compliance mandates.**

Policy Rule	TriggerCondition	IDSRespone
Deny access to unauthorized IP ranges	Access attempt from blacklisted IP	IP Block, Alert
Monitor all failed login attempts > 3	>3 failed SSH logins in 2 mins	Alert + Log Event
Stop data exfiltration attempts	Large data upload from user to unknown host	Connection Termination
Isolate infected systems	Malware signature detected	Host Quarantine
Inform admin about all suspicious activities	Any anomaly detection	Alert (Email + Dashboard)

Benefits of Mapping

- Enables **automated and consistent** responses.
- Supports **compliance** with standards (ISO 27001, NIST, GDPR).
- Reduces human error and ensures **timely mitigation**.
- Enhances **incident response planning.**

Vulnerability Analysis (Assessment)

Tests if a network/host is vulnerable to known attacks — complements IDS.

a special case of the intrusion detection process.

Process

- Sample specified system attributes.
- Store results securely.
- Compare with reference set (ideal config or previous snapshot).
- Identify & report differences.
- Optimizations: parallel engines, cryptographic integrity checks.

Vulnerability Analysis

Vulnerability Analysis (Assessment)- Types

- **By Location**
  - **Host-based** (Credential-based / Passive) – Uses internal data (files, configs, status). Best for privilege escalation detection.
  - **Network-based** (Non-credentialed / Active) – Remote probing or attack reenactment.
    - *Testing by exploit* – Simulate real attacks.
    - *Inference methods* – Look for indicators (version checks, open ports, protocol compliance).
- **By Trust Level**
  - Credentialed – With system access.
  - Non-credentialed – Without system access.

Vulnerability Analysis (Assessment)

Advantages

- Detects issues on systems that cannot support IDS.
- Documents & maintains system security baseline.
- Regular checks detect changes promptly.
- Validates that fixes don’t introduce new vulnerabilities.

Disadvantages / Issues

- Host-based: OS/application specific, costly to maintain.
- Network-based: Platform-independent but less accurate, more false alarms.
- Some tests can crash target systems.
- May interfere with IDS operations or train IDS to ignore real attacks.
- Must limit tests to authorized systems; address privacy concerns.

CREDENTIAL AND NON CREDENTIAL VULNERABILITY ANALYSIS

- **Credential vulnerability analysis** is performed **using valid user credentials** (such as admin or standard user logins) to **log into systems and perform in-depth internal scans.** This allows the scanner to access configuration settings, registry values, installed patches, and file systems, which are not visible from the outside.

Credential vulnerability analysis -Key Characteristics

- Requires authorized access to systems.
- Provides **deep insights** into system vulnerabilities.
- Can identify issues like:
  - Missing patches
  - Weak password policies
  - Misconfigured access rights
  - Outdated software versions
- More accurate, with fewer false positives.

Credential vulnerability analysis - Tools That Support It

- **Nessus** (with credentials)
- **Qualys** authenticated scans
- **OpenVAS** with login integration

Non-credential vulnerability analysis

- **Non-credential vulnerability analysis** (or unauthenticated scanning) is performed **without logging into the system.** It simulates the perspective of an external attacker attempting to find weaknesses through open ports, banner grabbing, and fingerprinting.

Non-Credential vulnerability analysis - Key Characteristics

- No login or access to internal system settings.
- Relies on network scans and open service responses.
- Can detect:
  - Open ports
  - Default credentials (based on banners)
  - Publicly known vulnerabilities (based on version info)
- Produces **more false positives**, but useful for external threat analysis.

Non-Credential vulnerability analysis - Tools That Support It

- **Nmap**
- **Nikto**
- **Nessus** unauthenticated scans
- **Shodan** (Internet-facing vulnerability data)

Feature	Credential Vulnerability Analysis	Non-Credential Vulnerability Analysis
Access Required	Yes – valid system credentials	No – works from outside without login
Depth of Analysis	Deep (internal configs, registry, files, patches)	Surface-level (network, open ports, services)
Accuracy	High – fewer false positives	Lower – may produce false alarms
Perspective	Internal (trusted user or admin)	External (attacker’s point of view)
Detects	Misconfigurations, missing patches, policy flaws	Exposed ports, known vulnerabilities, weak services
Use Case	Internal audits, compliance	Perimeter defense, external exposure assessment