Intrusion Detection Systems

(slides courtesy Prof. Stolfo)

Motivation

- We can't prevent all break-ins
- There will always be new holes, new attacks, and new attackers
- We need some way to cope

Defense in Depth

- More generically, most single defenses can fail
- We always need *defense in depth* – multiple layers, of different designs and philosophies
- One such layer: *Intrusion Detection Systems*

IDS Help

- An IDS alerted us to the sophisticated attack described last time
- We now know the machine had been penetrated at least as long ago as May
- But when the attacker tried to do more, he or she was detected – by an IDS

Just an Overview

- This is just a short overview of the subject
- For more details, take COMS E6185

Elements of Intrusion Detection

- Primary assumptions:
  - System activities are observable
  - Normal and intrusive activities have distinct evidence
- Components of intrusion detection systems:
  - From an algorithmic perspective:
    - Features - capture intrusion evidence from audit data
    - Models - piece evidence together; infer attack
  - From a system architecture perspective:
    - Audit data processor, knowledge base, decision engine, alarm generation and responses

Host-Based IDSs

- Using OS auditing mechanisms
  - E.G., BSM on Solaris: logs all direct or indirect events generated by a user
  - *strace* for system calls made by a program
- Monitoring user activities
  - E.G., Analyze shell commands
- Monitoring execution of system programs
  - E.G., Analyze system calls made by *sendmail*

Basic Audit Modules (Hosts)

Windows Registry sensor

EventLog - Uses the windows Event Logging system to track entries into all three of the windows event logs: System, Security, Application

Netstat - Uses the information from the program *netstat* to provide information about network usage on the machine

Health - Runs the program *health* to give current information about the system (CPU usage, mem usage, swap usage)

Ps - Uses information from the /proc virtual file system as a data source

System Call Traces

- [pid 1286] execve 11:33:27;[pid 1286] open 11:33:27;[pid 1286] mmap 11:33:27;[pid 1286] open 11:33:27;[pid 1286] mmap 11:33:27;[pid 1286] mmap 11:33:27;[pid 1286] munmap 11:33:27;[pid 1286] mmap 11:33:27;[pid 1286] mmap 11:33:27;[pid 1286] close 11:33:27;[pid 1286] open 11:33:27;[pid 1286] mmap 11:33:27;[pid 1286] mmap 11:33:27;[pid 1286] munmap 11:33:27;[pid 1286] mmap 11:33:27;[pid 1286] close 11:33:27;[pid 1286] open 11:33:27;[pid 1286] mmap 11:33:27;[pid 1286] mmap 11:33:27;[pid 1286] munmap 11:33:27;[pid 1286] mmap 11:33:27;[pid 1286] mmap 11:33:27;[pid 1286] close 11:33:27;[pid 1286] open 11:33:27;[pid 1286] mmap 11:33:27;[pid 1286] close 11:33:27;[pid 1286] open 11:33:27;[pid 1286] mmap 11:33:27;[pid 1286] mmap 11:33:27;[pid 1286] munmap 11:33:27;[pid 1286] mmap 11:33:27;[pid 1286] close 11:33:27;[pid 1286] open 11:33:27;[pid 1286] mmap 11:33:27;[pid 1286] mmap 11:33:27;[pid 1286] munmap 11:33:27;[pid 1286] mmap 11:33:27;[pid 1286] close 11:33:27;[pid 1286] open 11:33:27;[pid 1286] mmap 11:33:27;[pid 1286] mmap 11:33:27;[pid 1286] munmap 11:33:27;[pid 1286] mmap 11:33:27;[pid 1286] mmap 11:33:27;[pid 1286] close 11:33:27;[pid 1286] open 11:33:27;[pid 1286] mmap 11:33:27;[pid 1286] mmap 11:33:27;[pid 1286] munmap 11:33:27;[pid 1286] mmap 11:33:27;[pid 1286] close 11:33:27;[pid 1286] open 11:33:27;[pid 1286] mmap 11:33:27;[pid 1286] mmap 11:33:27;[pid 1286] munmap 11:33:27;[pid 1286] mmap 11:33:27;[pid 1286] close 11:33:27;[pid 1286] open 11:33:27;[pid 1286] mmap 11:33:27;[pid 1286] mmap 11:33:27;[pid 1286] munmap 11:33:27;[pid 1286] mmap 11:33:27;[pid 1286] close 11:33:27;[pid 1286] close 11:33:27;[pid 1286] munmap 11:33:27;[pid 1286] open 11:33:27;[pid 1286] ioctl 11:33:27;[pid 1286] close 11:33:27;[pid 1286] nice 11:33:27;[pid 1286] auditon 11:33:27;[pid 1286] open 11:33:27;[pid 1286] ioctl 11:33:27;[pid 1286] close 11:33:27;[pid 1286] open 11:33:27;[pid 1286] ioctl

Windows Registry Accesses

```
Smmc.exe SOpenKey
SHKLM\Software\Microsoft\Windows_NT\CurrentVersion\FontLink\SystemLink
SNOTFOUND S0     NORMAL_
Smmc.exe SOpenKey
SHKLM\Software\Microsoft\Windows_NT\CurrentVersion\FontLink\SystemLink
SNOTFOUND S0     NORMAL_
SREGMON.EXE SOpenKey
SHKLM\System\CurrentControlSet\Services\WinSock2\Parameters SSUCCESS
SKey:_0xE12F4580 NORMAL_
SREGMON.EXE SQueryValue
SHKLM\System\CurrentControlSet\Services\WinSock2\Parameters\WinSock_Regi
stry_Version SSUCCESS S"2.0"       NORMAL_
SREGMON.EXE SQueryValue
SHKLM\System\CurrentControlSet\Services\WinSock2\Parameters\WinSock_Regi
stry_Version SSUCCESS S"2.0"       NORMAL_
SREGMON.EXE SOpenKey
SHKLM\System\CurrentControlSet\Services\WinSock2\Parameters\Protocol_Cat
alog9 SSUCCESS SKey:_0xE1F07580    NORMAL_
SREGMON.EXE SQueryValue
SHKLM\System\CurrentControlSet\Services\WinSock2\Parameters\Protocol_Cat
alog9\Serial_Access_Num SSUCCESS S0x4       NORMAL_
```

Network IDSs

- Deploying sensors at strategic locations
  - E.G., Packet sniffing via *tcpdump* at routers
- Inspecting network traffic
  - Watch for violations of protocols and unusual connection patterns
- Monitoring user activities
  - Look into the data portions of the packets for malicious command sequences
- May be easily defeated by encryption
  - Data portions and some header information can be encrypted
- Other problems …

Network Connections

0,tcp,http,SF,181,5450,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,8,8,0.00,0.00,0.00,0.00,1.00,0.00,0.00,9,9,1.00,0.00,0.11,0.00,0.00,0.00,0.00,0.00,normal.

0,tcp,http,SF,239,486,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,8,8,0.00,0.00,0.00,0.00,1.00,0.00,0.00,19,19,1.00,0.00,0.05,0.00,0.00,0.00,0.00,0.00,normal.

0,tcp,http,SF,235,1337,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,8,8,0.00,0.00,0.00,0.00,1.00,0.00,0.00,29,29,1.00,0.00,0.03,0.00,0.00,0.00,0.00,0.00,normal.

0,tcp,http,SF,219,1337,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,6,6,0.00,0.00,0.00,0.00,1.00,0.00,0.00,39,39,1.00,0.00,0.03,0.00,0.00,0.00,0.00,0.00,normal.

0,tcp,http,SF,217,2032,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,6,6,0.00,0.00,0.00,0.00,1.00,0.00,0.00,49,49,1.00,0.00,0.02,0.00,0.00,0.00,0.00,0.00,normal.

0,tcp,http,SF,217,2032,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,6,6,0.00,0.00,0.00,0.00,1.00,0.00,0.00,59,59,1.00,0.00,0.02,0.00,0.00,0.00,0.00,0.00,normal.

0,tcp,http,SF,212,1940,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,1,2,0.00,0.00,0.00,0.00,1.00,0.00,1.00,1,69,1.00,0.00,1.00,0.04,0.00,0.00,0.00,0.00,normal.

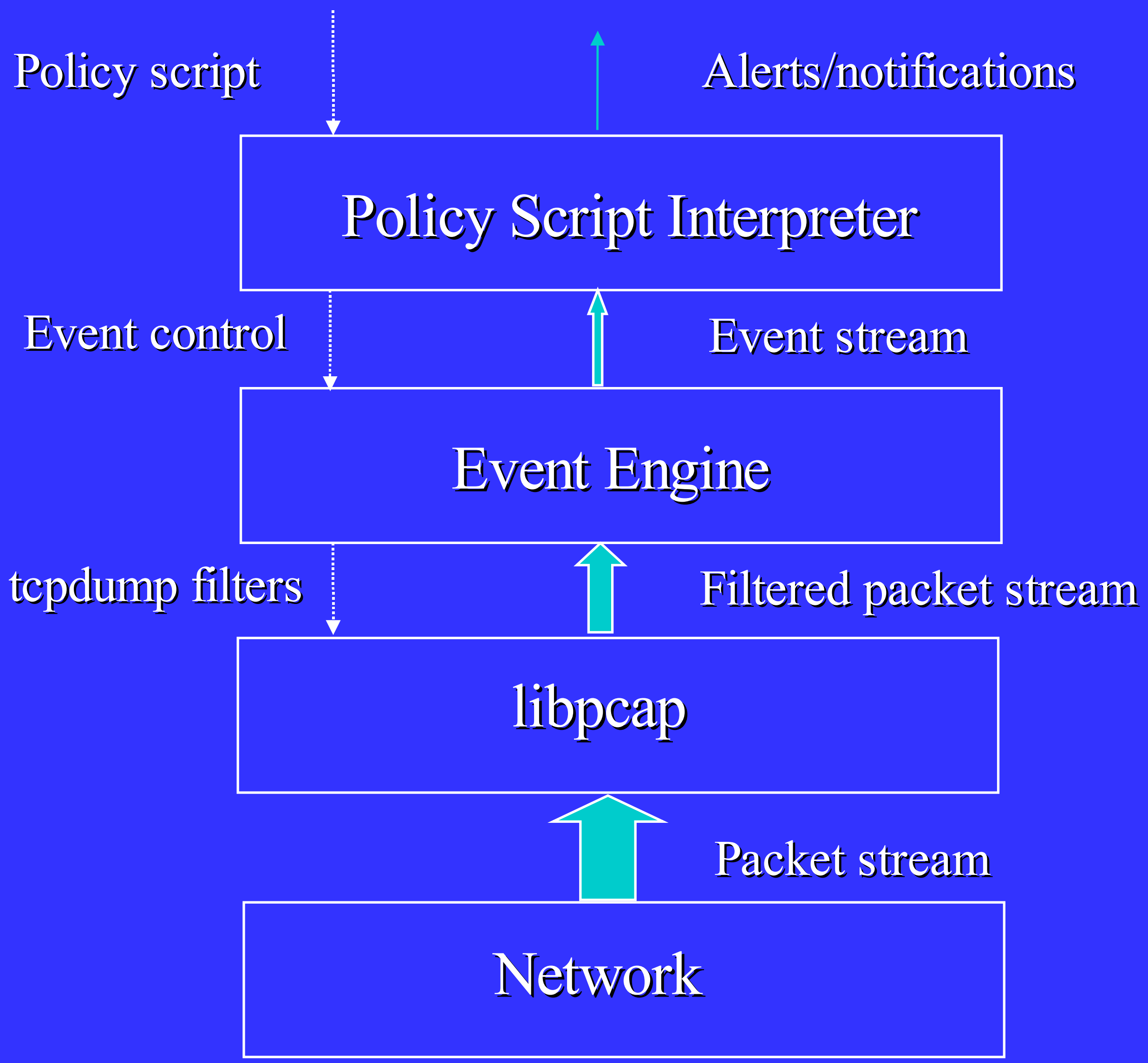
0,tcp,http,SF,159,4087,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,5,5,0.00,0.00,0.00,0.00,1.00,0.00,0.00,11,79,1.00,0.00,0.09,0.04,0.00,0.00,0.00,0.00,normal.

0,tcp,http,SF,210,151,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,8,8,0.00,0.00,0.00,0.00,1.00,0.00,0.00,8,89,1.00,0.00,0.12,0.04,0.00,0.00,0.00,0.00,normal.

0,tcp,http,SF,212,786,0,0,0,1,0,1,0,0,0,0,0,0,0,0,0,0,8,8,0.00,0.00,0.00,0.00,1.00,0.00,0.00,8,99,1.00,0.00,0.12,0.05,0.00,0.00,0.00,0.00,attack.

0,tcp,http,SF,210,624,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,18,18,0.00,0.00,0.00,0.00,1.00,0.00,0.00,18,109,1.00,0.00,0.06,0.05,0.00,0.00,0.00,0.00,normal.

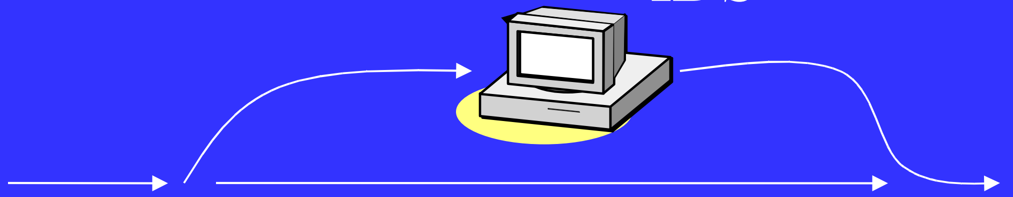
Architecture of Network IDS

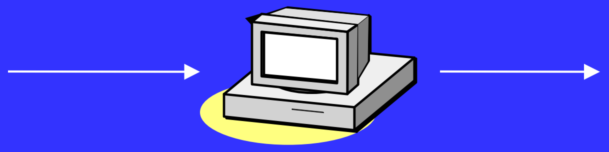


Firewall Versus Network IDS

- Firewall
  - Active filtering
  - Fail-close
- Network IDS
  - Passive monitoring
  - Fail-open

IDS





FW

Requirements of Network IDS

- High-speed, large volume monitoring
  - No packet filter drops
- Real-time notification
- Mechanism separate from policy
- Extensible
- Broad detection coverage
- Economy in resource usage
- Resilience to stress
- Resilience to attacks upon the IDS itself!

Eluding Network IDS

- What the IDS sees may not be what the end system gets.
  - Insertion and evasion attacks.
    - IDS needs to perform full reassembly of packets.
  - But there are still ambiguities in protocols and operating systems:
    - E.G. TTL, fragments.
    - Need to "normalize" the packets.

Insertion Attack

End-System sees:

A	T	T	A	C	K

IDS sees:

A	T	X	T	A	C	K

Attacker’s data stream

T	X	T	C	A	A	K

Examples: bad checksum, TTL.

Evasion Attack

End-System sees:

A T T A C K

IDS sees:

A T T C K

Attacker's data stream

T T C A A K

Example: fragmentation overlap

DoS Attacks on Network IDS

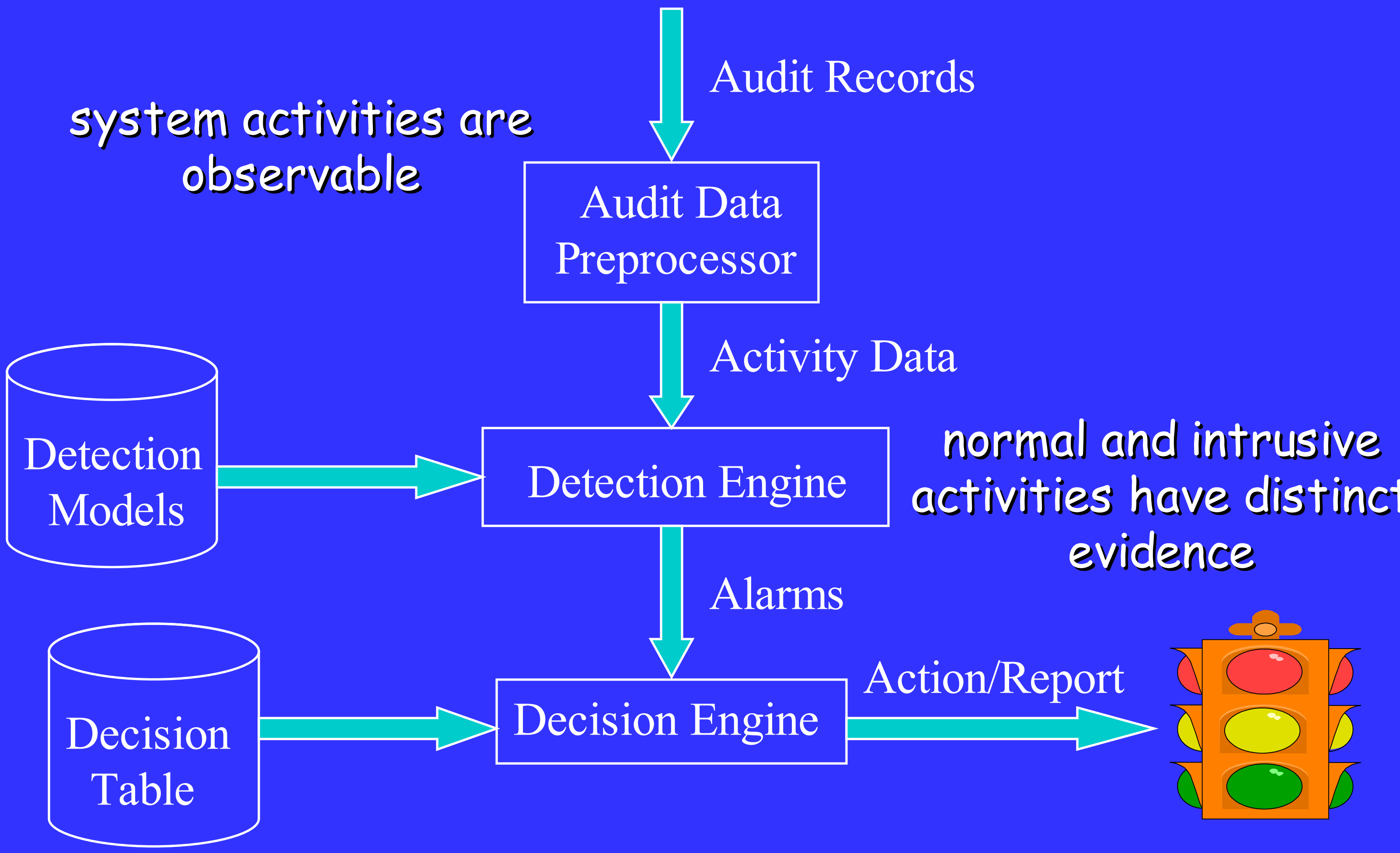
- Resource exhaustion
  - CPU resources
  - Memory
  - Network bandwidth
- Abusing reactive IDS
  - False positives
  - Nuisance attacks or “error” packets/connections

Taxonomy of IDS's

Intrusion Detection Approaches

- Modeling
  - Features: evidences extracted from audit data
  - Analysis approach: piecing the evidences together
    - Misuse detection (a.k.a. signature-based)
    - Anomaly detection (a.k.a. statistical-based)
- Deployment: Network-based or Host-based
- Development and maintenance
  - Hand-coding of “expert knowledge”
  - Learning based on audit data

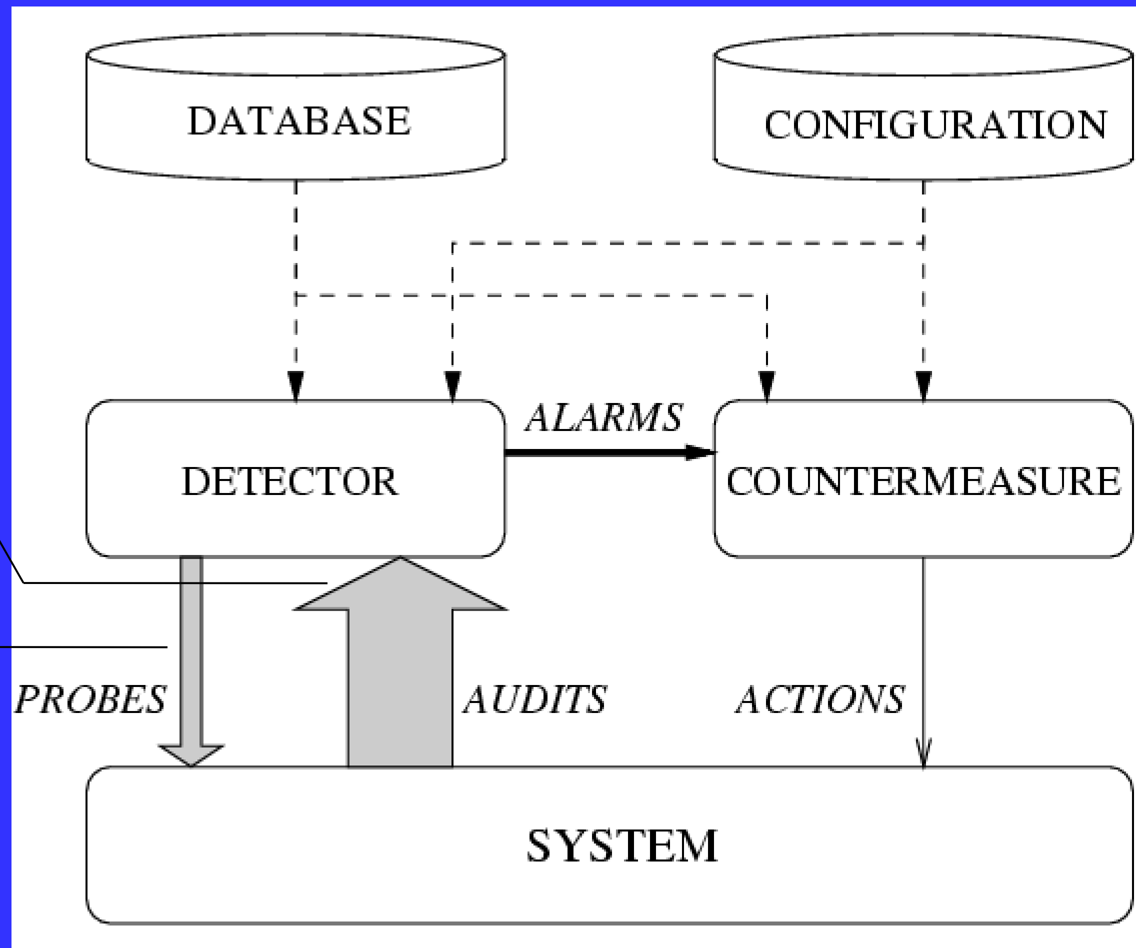
Components of Intrusion Detection System



A Generic IDS

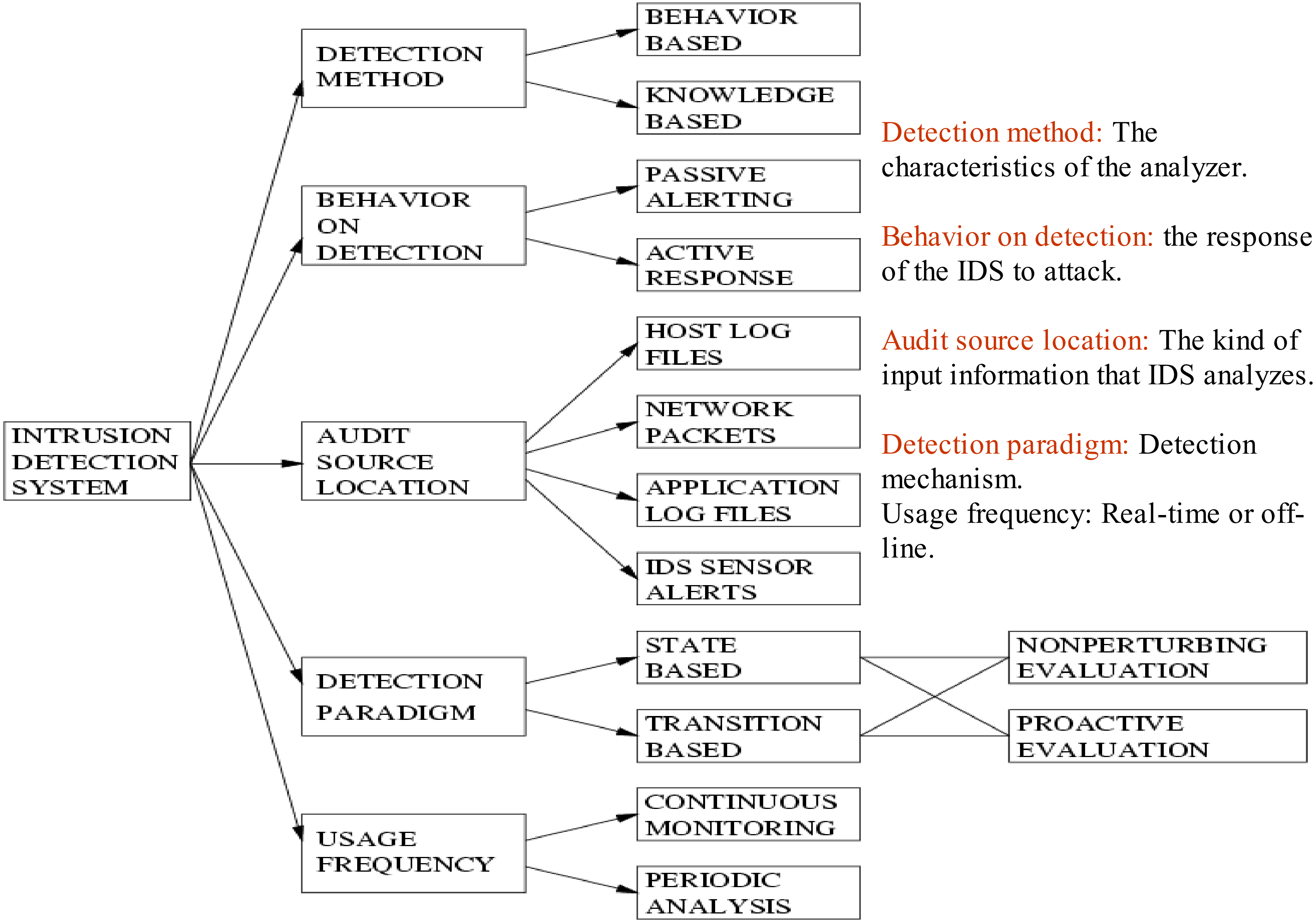
Information provided by a system concerning its inner workings and behavior

System Vulnerability Analysis Port-scanning, etc.



Detector: Eliminates unneeded information from the audit trail.
Countermeasure: Takes corrective action to either prevent the actions from being executed or changing the state of the system back to a secure state.

Characteristics of IDS



Detection method: The characteristics of the analyzer.

Behavior on detection: the response of the IDS to attack.

Audit source location: The kind of input information that IDS analyzes.

Detection paradigm: Detection mechanism.
Usage frequency: Real-time or off-line.

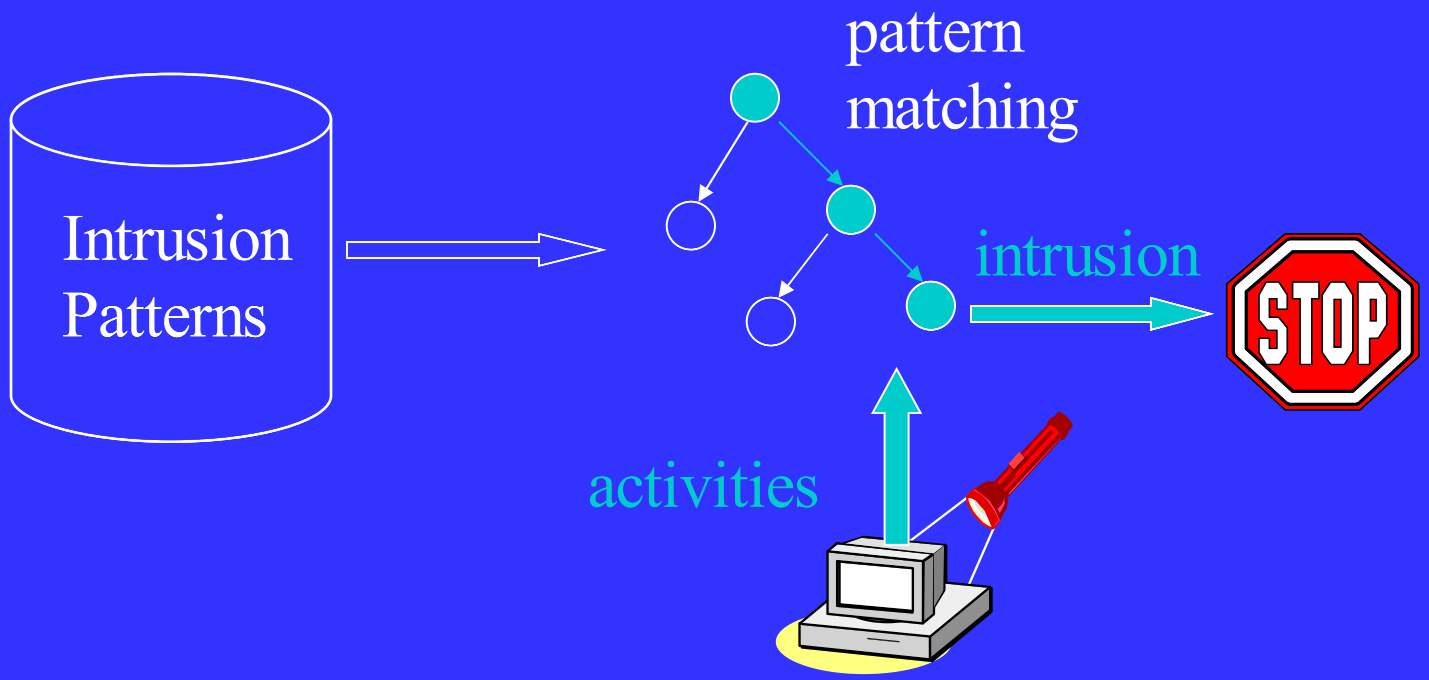
Detection Paradigm

- State-based versus transition-based IDS
  - State-based: Identifies intrusions on the states
  - Transition-based: Watches events that trigger transition from one state to another
- Non-perturbing versus pro-active analysis of state or transition
  - Non-perturbing: Consists of the vulnerability assessment side
  - Pro-active: Analysis by explicitly triggering events

IDS: Time aspect

- Real-time IDS
  - Analyzes the data while the sessions are in progress
  - Raises an alarm immediately when the attack is detected
- Off-line IDS
  - Analyzes the data after the information has been already collected
  - Useful for understanding the attackers' behavior

Misuse Detection



Example: ***if*** (src_ip == dst_ip) ***then*** "land attack"

Can't detect new attacks

Misuse Detection

- The system is equipped with a number of attack descriptions ("signature"). Then matched against the audit data to detect attacks.
- Pro: less false positives (But there still some!)
- Con: cannot detect novel attacks, need to update the signatures often.
- Approaches: pattern matching, security rule specification.

Knowledge-based IDS

- Good accuracy, bad completeness
- Drawback: need regular update of knowledge
  - Difficulty of gathering the information
  - Maintenance of the knowledge is a time-consuming task
- Knowledge-based IDS
  - Expert systems
  - Signature analysis
  - Petri nets
  - State-transition analysis

Specification-based Detection

- Manually develop specifications that capture legitimate (not only previous seen) system behavior. Any deviation from it is an attack
- Pro: can avoid false-positive since the specification can capture all legitimate behavior.
- Con: hard to develop a complete and detailed specification, and error-prone.
- Approach: state machine, extended finite state automata (EFSA)
  - Augment FSA with state variables
  - Make transition on event that may have arguments

Example of specification-based IDS

A gateway's behavior at IP

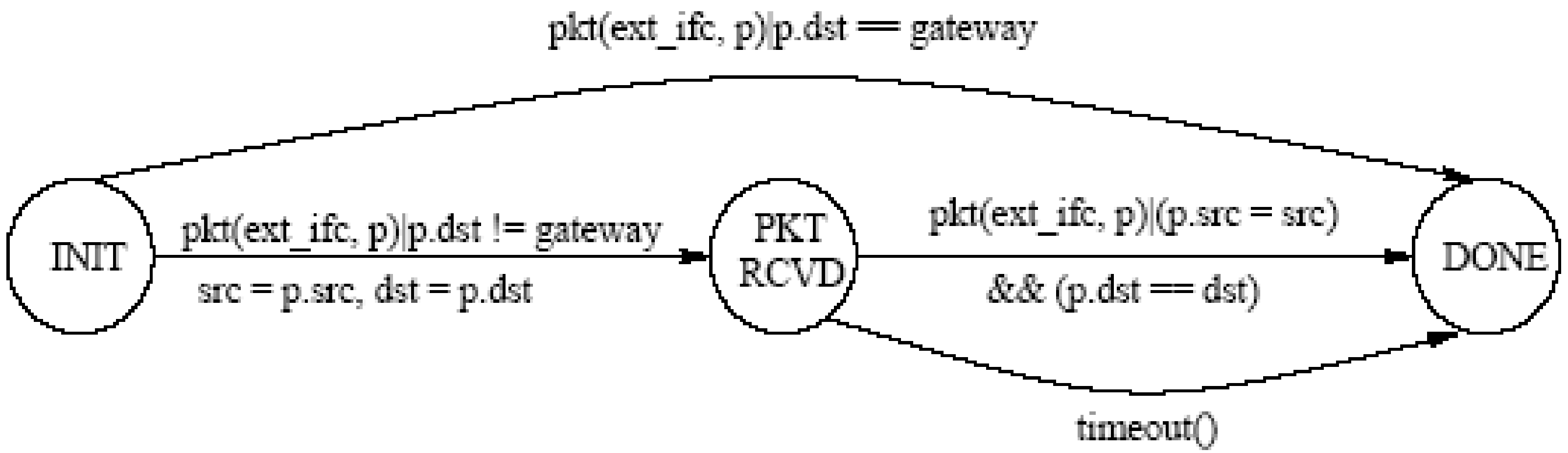


Figure 1: Simplified IP Protocol State Machine

State variables: src, dst. Event: pkt(ext_ifc, p), timeout.

ext_ifc is the network interface on which packet received, and p is the packet content

Today's IT Security Tools

- We make lists of bad behavior
  - Virus definitions
  - SPAM filters and blacklists
  - IDS signatures
  - Policies
- We distribute the lists to applications and detection systems
- They flag behavior that fits the pattern
- The system is about to collapse
  - Delays
  - Administrative Overhead
  - False positives

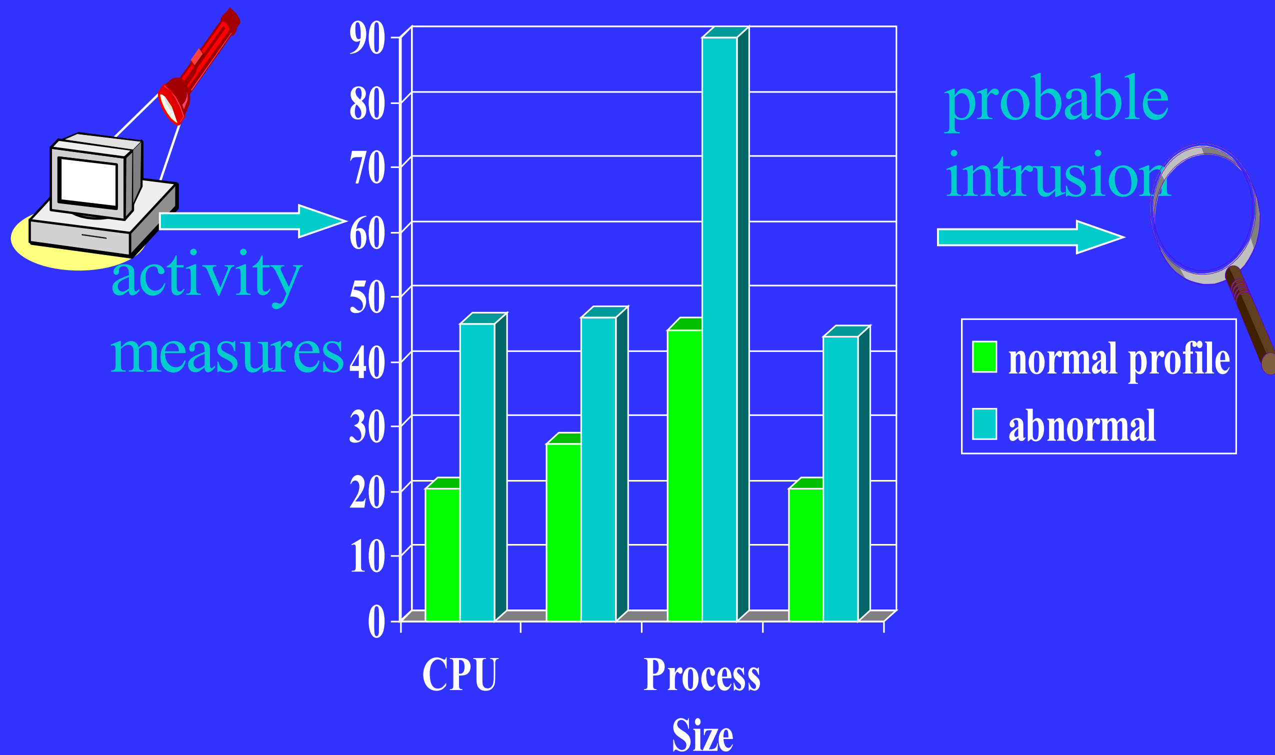
Behavior-based IDS

- Good completeness, bad accuracy
- Detect intrusion by observing a deviation from the normal or expected behavior of the system or the users
- Can detect attempts to exploit new and unforeseen vulnerabilities
- Behavior-based IDS
  - Statistics
  - Expert systems
  - Neural networks
  - User intention identification
  - Computer immunology

Anomaly Detection

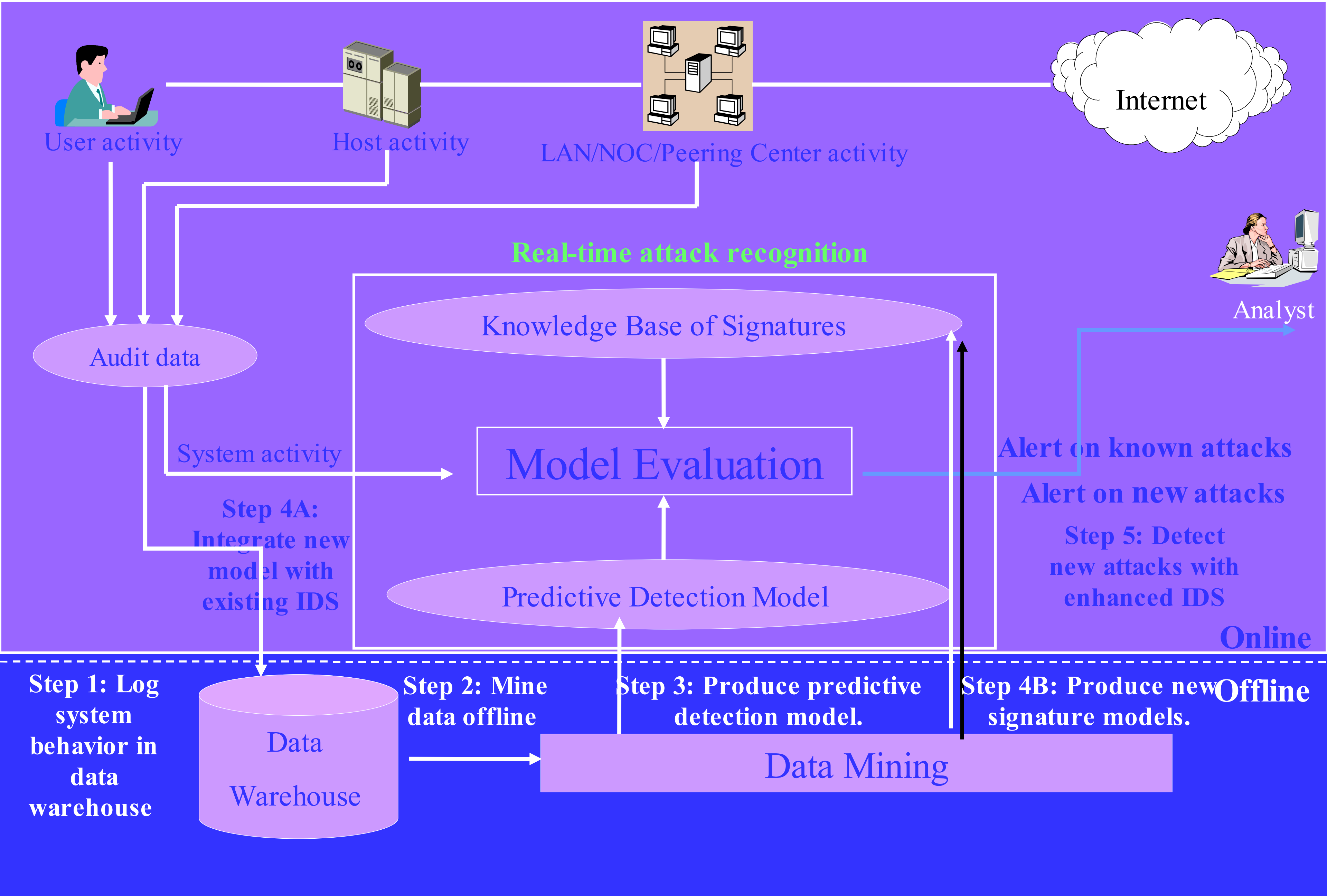
- Build models of “normal” behavior of a system using machine learning or data mining. Any large deviation from the model is thought as anomaly.
- Pro: can detect previous unseen attacks
- Con: have higher false positives, and hard to train a system for a very dynamic environment.
- Approaches: statistical methods, clustering, outlier detection, SVM

Anomaly Detection



Relatively high false positive rate -
anomalies can just be new normal activities.

Data Mining System Perspective



Anomaly Detection

- Model
  - Generative / Discriminative
- Algorithm
  - Supervised / unsupervised
  - Compute online?
- Data source / feature selection
  - Depends on expert knowledge now
- Cost
  - Computation cost
  - Feature audit and construction cost
  - Damage cost
- Goal: detect attacks accurately and promptly

Data sources

- Single packet
  - src and dst ip, port (most commonly used)
  - All packet header fields (PHAD)
- A sequence of packets
  - Follow the automaton for the protocols (specification-based)
- Reconstructed connections
  - Connection status, frequency (commonly used)
- Application data
  - Character distribution, keywords, etc. (ALAD, www ids)
- Traffic flows
  - Volume / velocity. (signal analysis, k-ary sketch, PCAP)

Supervised Learning

- Statistical tests
  - Build distribution model for normal behavior, then detect low probability events
- Outlier detection
  - K-Nearest neighbor, Mahalanobis distance, LOF
- Self-Organizing Map (SOM) [Ramadas 03]
- Nonstationary model - PHAD/ALAD [Mahoney 02]
- Probability AD (PAD) [Stolfo, Eskin 04]
- SVM / OCSVM

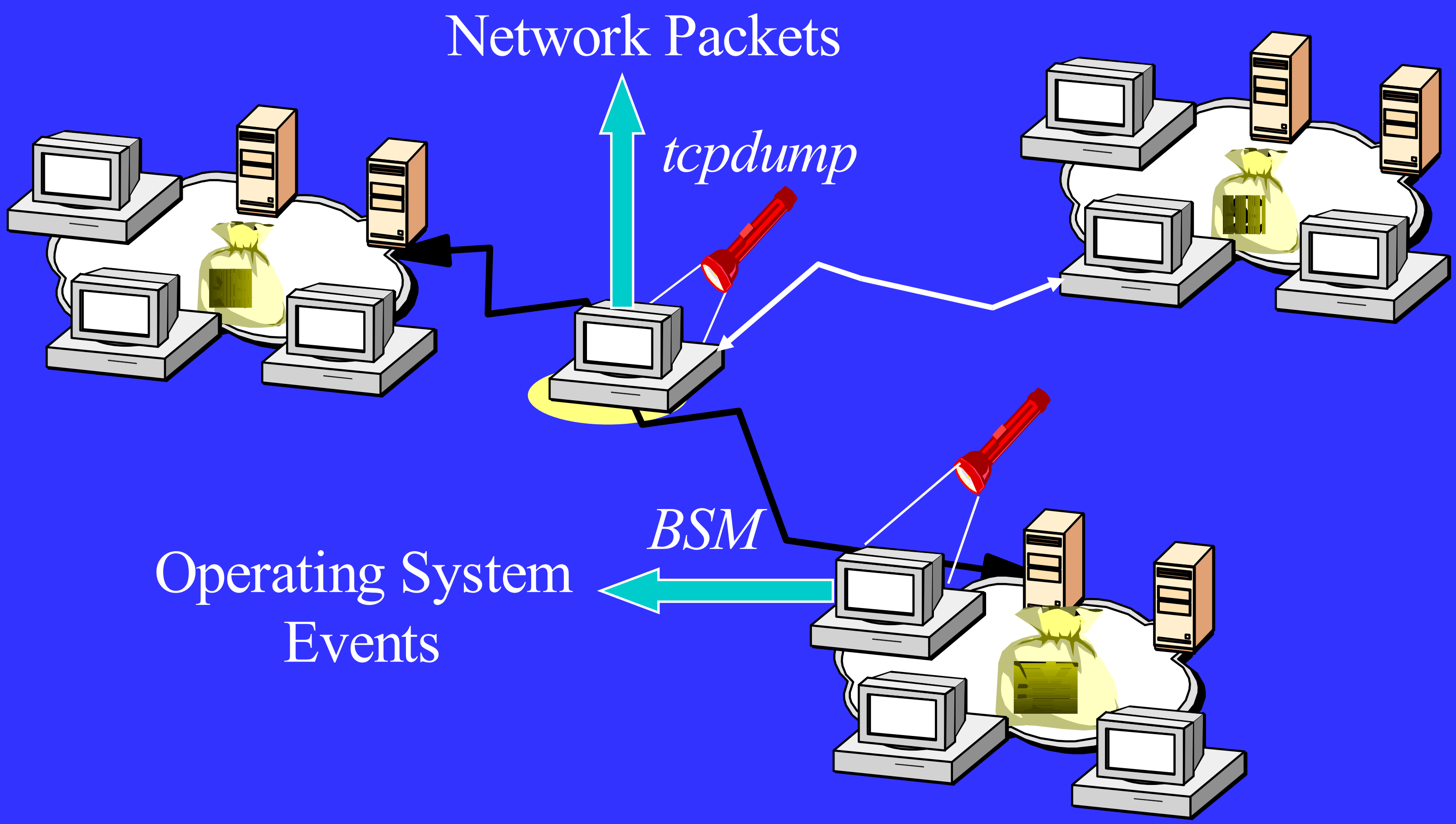
Unsupervised Learning

- Outlier detection
- Clustering
- SmartSifter [Yamanishi 00]
  - Online learning
  - Histogram + Finite mixtures
- Wavelet analysis for change detection [Barford 02]
- OCSVM
- Most of them cannot used for real-time detection

Examples of IDS

- Misuse detection
  - SNORT: signature based commercial IDS
  - STAT: real time IDS using state transition analysis, attack scenarios specified by STATL. (Higher level signature, abstract from raw packet) [Vigna 03]
  - Bro: real time, events driven, security policy written in a specialized script language. [Paxson 99]
- Anomaly detection
  - MADAM ID : use RIPPER
  - ADAM: mining association rule + Bayes classifier
- Specification-based detection [Sekar 02]

Hybrid NIDS and HIDS



Host-based Information Sources

- Must be real-time
- System sources
  - Commands of Operating Systems don’t offer a structural way of collecting and storing the audit information
- Accounting: Shared resources
  - Untrustworthy for security purposes
  - Syslog
- C2 security audit
  - Reliable
  - Trusted Computing Base (TCB)

Network-based information sources

- Simple Network Management Protocol (SNMP) Management Information Base (MIB)
  - A repository of information
- Network packets
  - Detection of network-specific attacks
  - Can analyze the payload of the packet
- Router NetFlow records
  - Can speed up and create log

Evaluation of IDS

- Accuracy
  - Detection rate & false alarm
- Performance
- Completeness
  - To predict new attacks
- Fault tolerance
- Timeliness

Key Performance Metrics

- Algorithm
  - Alarm: A; Intrusion: I
  - Detection (true alarm) rate: $P(A|I)$
    - False negative rate $P(\neg A|I)$
  - False alarm rate: $P(A|\neg I)$
    - True negative rate $P(\neg A|\neg I)$
  - Bayesian detection rate: $P(I|A)$
- Architecture
  - Scalable
  - Resilient to attacks

Bayesian Detection Rate

$$P(I \mid A) = \frac{P(I)P(A \mid I)}{P(I)P(A \mid I) + P(\neg I)P(A \mid \neg I)}$$

- Base-rate fallacy
  - Even if false alarm rate $P(A|\neg I)$ is very low, Bayesian detection rate $P(I|A)$ is still low if base-rate $P(I)$ is low
  - E.g. if $P(A|I) = 1$, $P(A|\neg I) = 10^{-5}$, $P(I) = 2\times10^{-5}$, $P(I|A) = 66\%$
- Implications to IDS
  - Design algorithms to reduce false alarm rate
  - Deploy IDS to appropriate point/layer with sufficiently high base rate

Problems with (Commercial) IDS

- Cost of update and keeping current is growing
  - Organizations lack internal expertise
  - MSSP industry also suffering
- IDS systems suffer from False Negative Problem
  - New augmented IDS with Anomaly Detectors are appearing in the commercial market
  - Initial focus on protocols
- IDS are inherently noisy and chatty and suffer from the False Positive problem
  - Volumes of alerts are crushing
  - Honing in on most serious threats is hard
- NIDS positioned at the perimeter
  - The most serious/predominant threat is the insider
  - Host and LAN-based IDS now more crucial

What new solutions are needed for these problems?

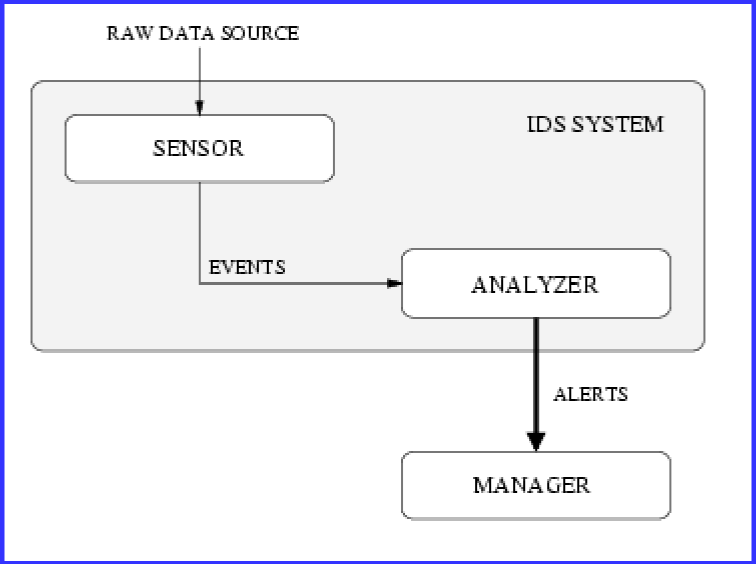
- Maintenance problem – Automatic Update
- Limited coverage problem – False Negative/Zero Day
- Data Reduction problem – Human can’t be in the loop
- Insider problem – Look inward, not only outward

Next Generation Detection Systems

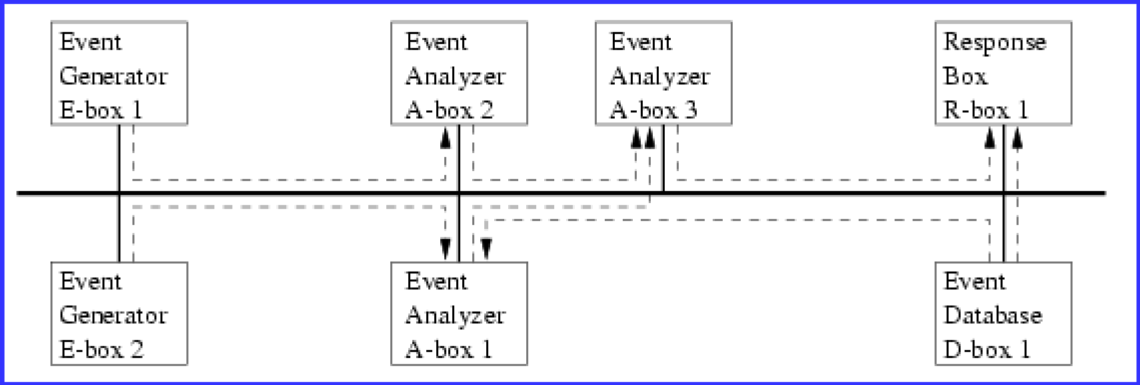
- Behavior-based (like credit card fraud):
  - Automated analysis
  - Learn site specific characteristics (e.g., outbound traffic) and prioritize attacks per cost modeling
  - Reduce time to update and deploy
  - Increase analyst/security staff productivity
  - <u>Discover New Attacks</u>
- Offload and load balance detection tasks among separate specialized modules
- Correlation among distributed sites provides new opportunities for
  - Real-time global detection (early warning)
  - Detecting attackers (deterrent)

The Reusability Issue

Intrusion Detection exchange format Working Group (IDWG): Address the problem of communication between IDS and external components.



Common Intrusion-Detection Framework (CIDF): Coordinate different IDS projects.

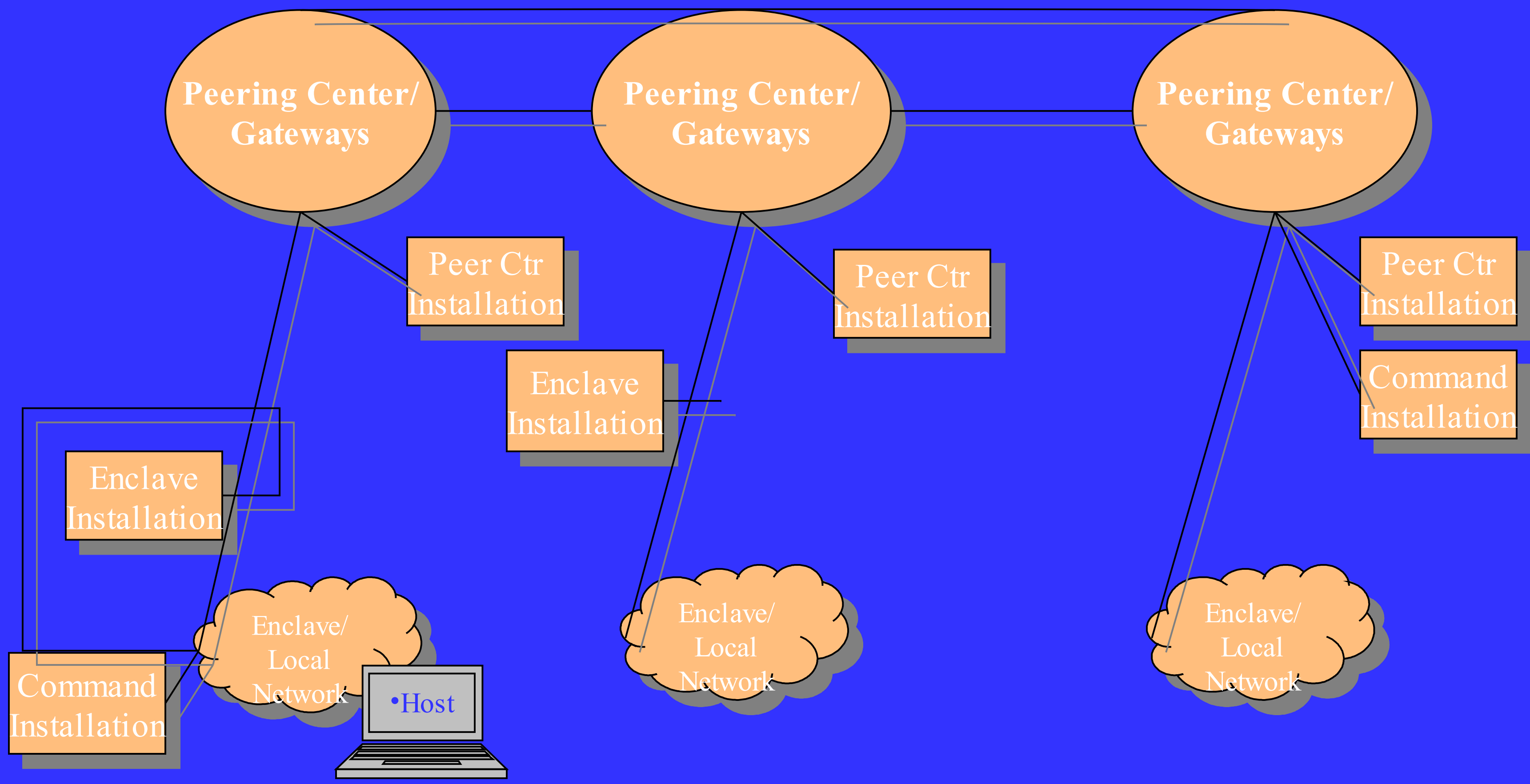


Paradigm Shift

IN IDS

Signature-Based	Defense Strategy →	Behavior-Based
Human Expertise	Data Analysis →	Machine Expertise
Generic	System Architecture →	Specific
Fragmented	Coverage →	Distributed Cooperative
Attacks	Detection →	Attacker

Collaborative Network Architecture



Provide information assurance through real-time sharing technology in a distributed, scalable and coordinated environment