Intrusion Detection Systems

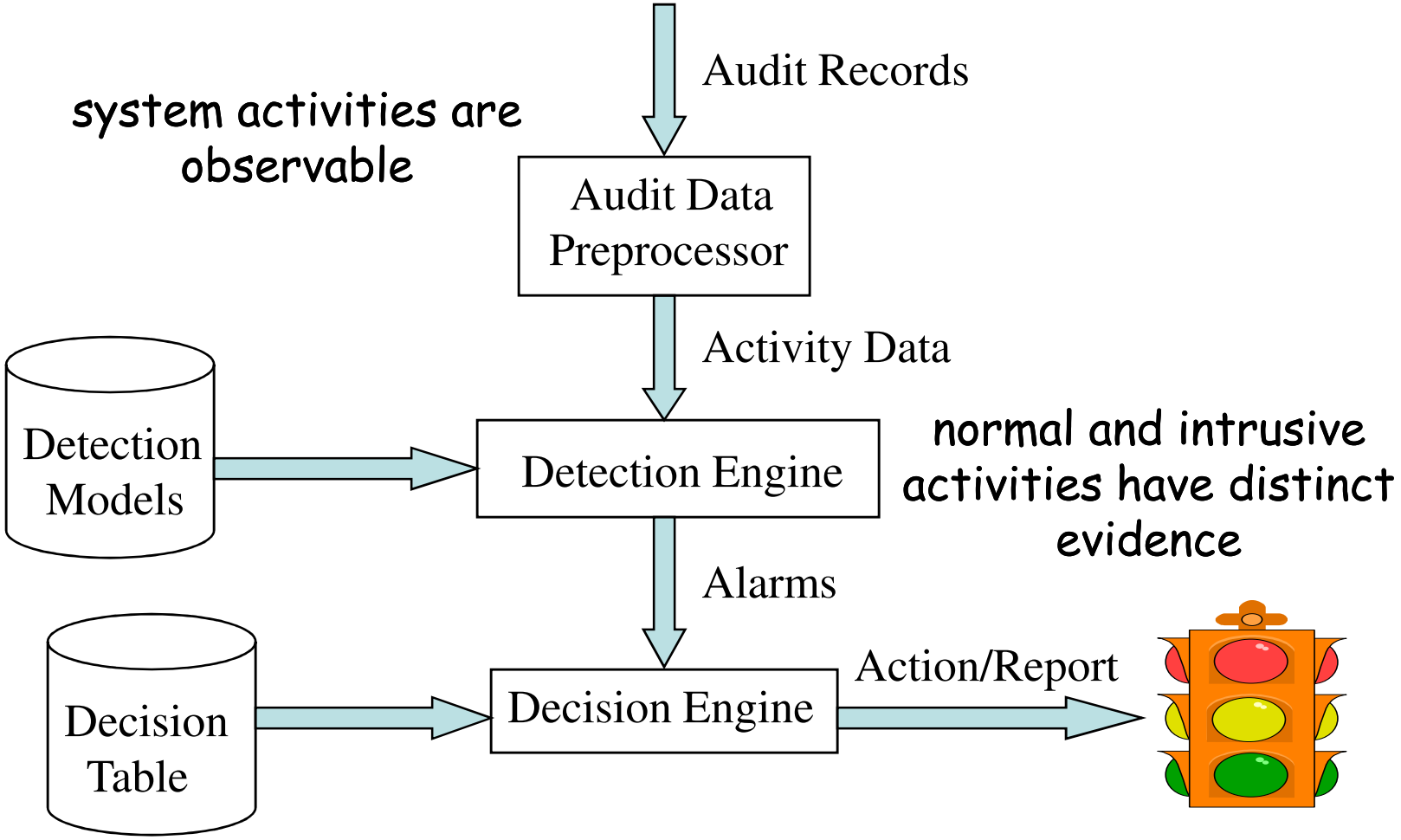
Definitions

- Intrusion
  - A set of actions aimed to compromise the security goals, namely
    - Integrity, confidentiality, or availability, of a computing and networking resource
- Intrusion detection
  - The process of identifying and responding to intrusion activities

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 evidences
    - Models - piece evidences together
  - From a system architecture perspective:
    - Audit data processor, knowledge base, decision engine, alarm generation and responses

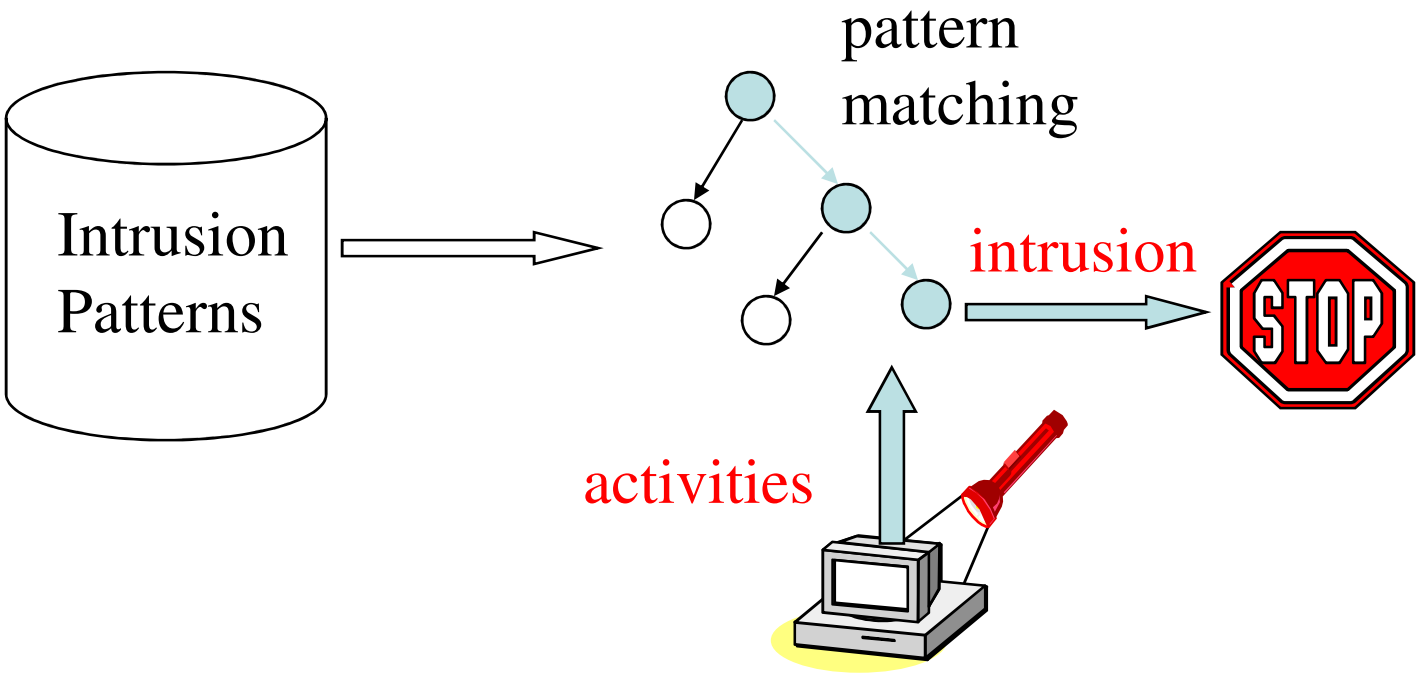
Components of Intrusion Detection System



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

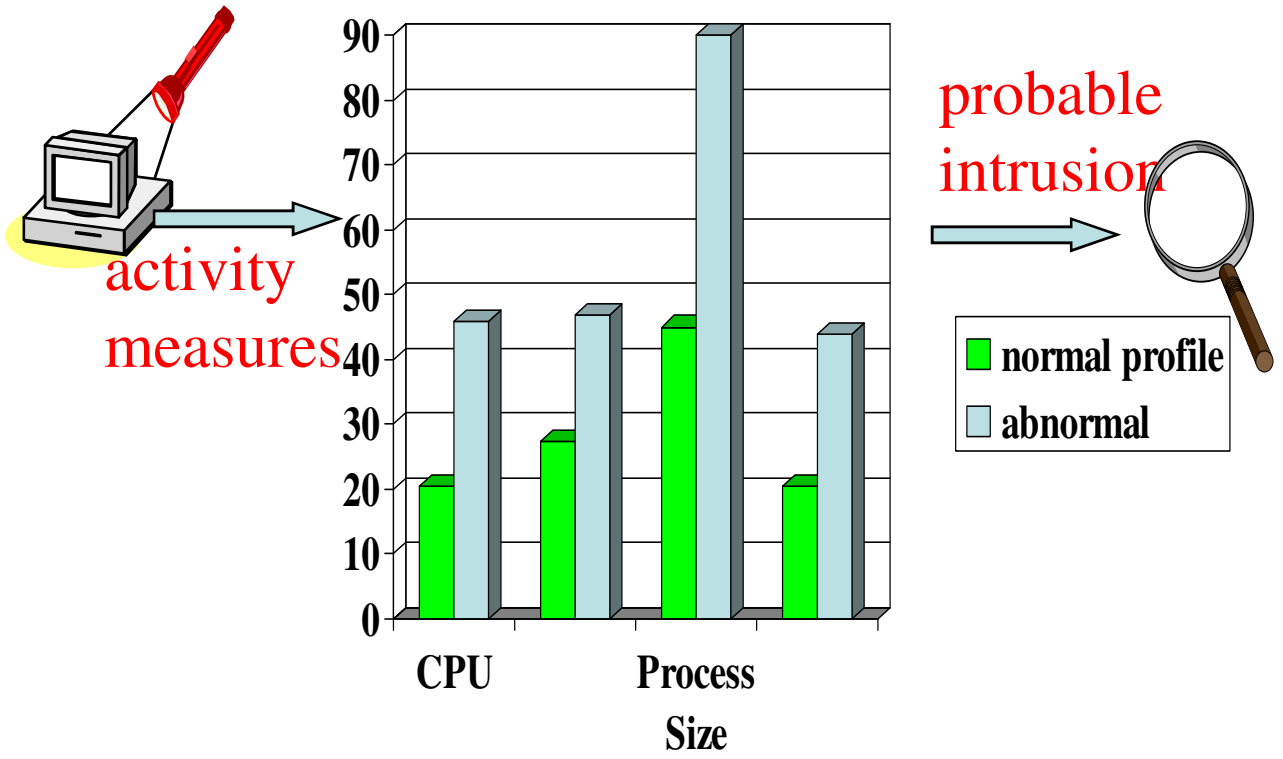
Misuse Detection



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

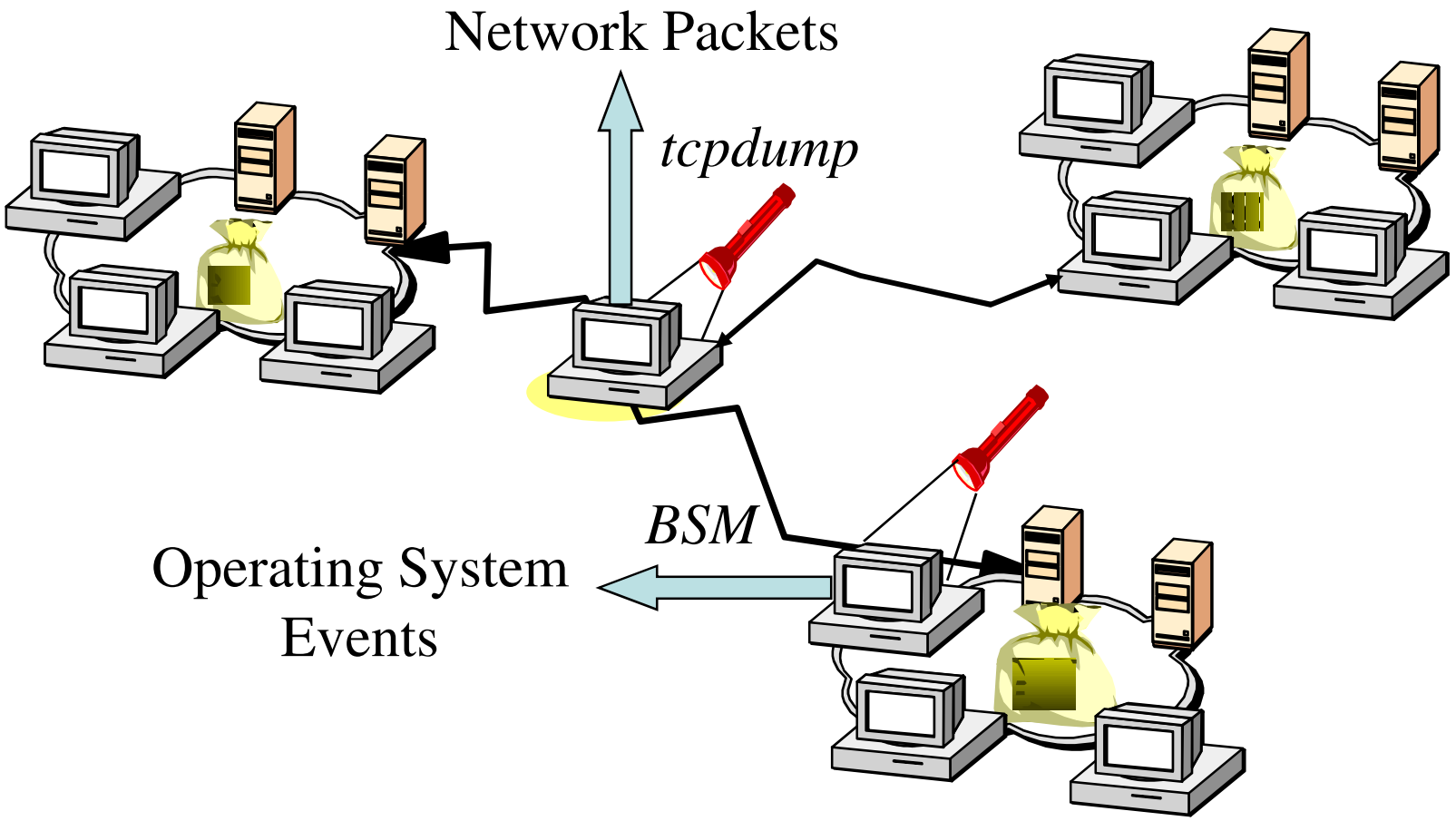
Can't detect new attacks

Anomaly Detection



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

Monitoring Networks and Hosts



Key Performance Metrics

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

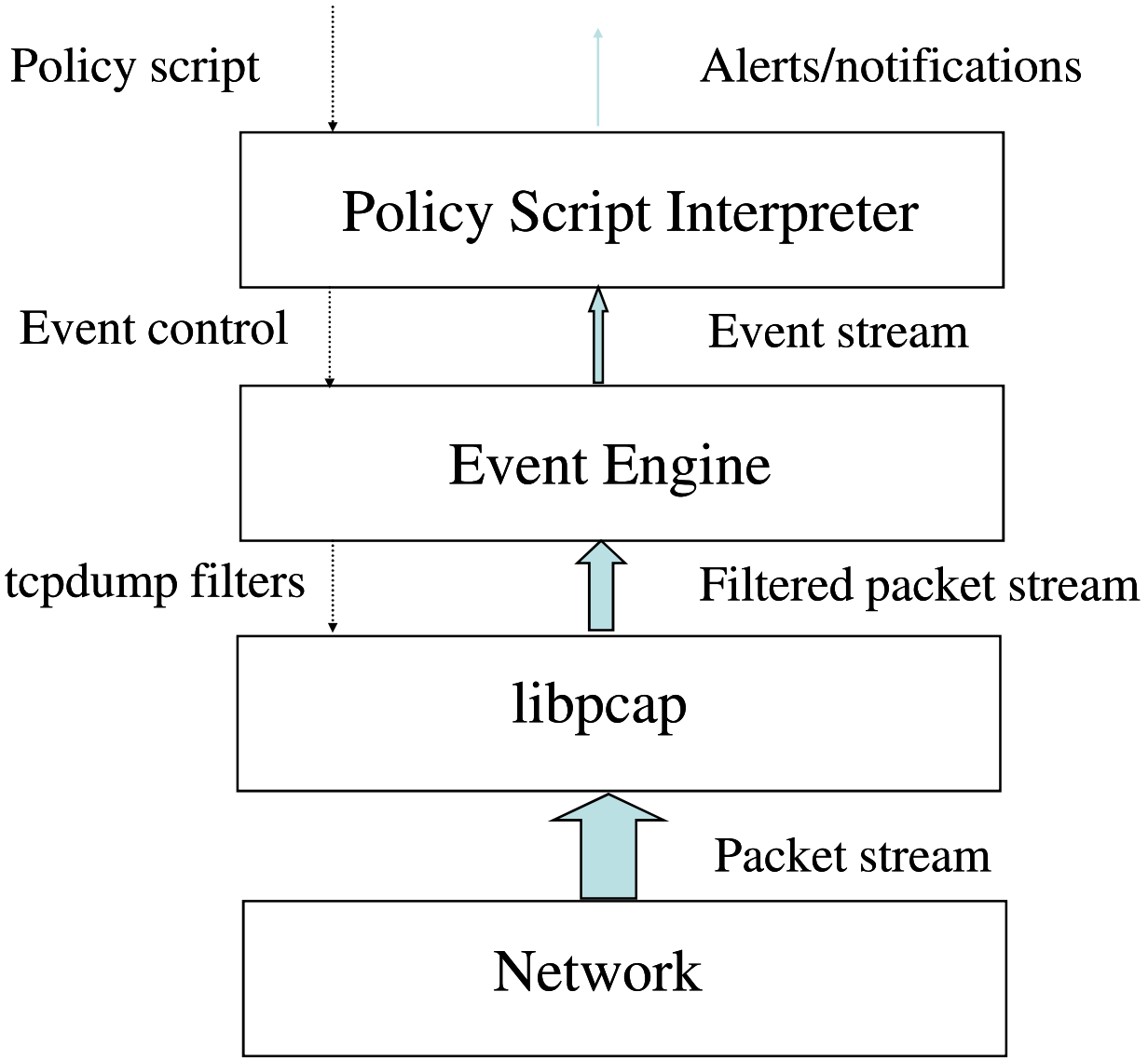
Host-Based IDSs

- 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 executions of system programs
  - E.G., Analyze system calls made by *sendmail*

Network IDSs

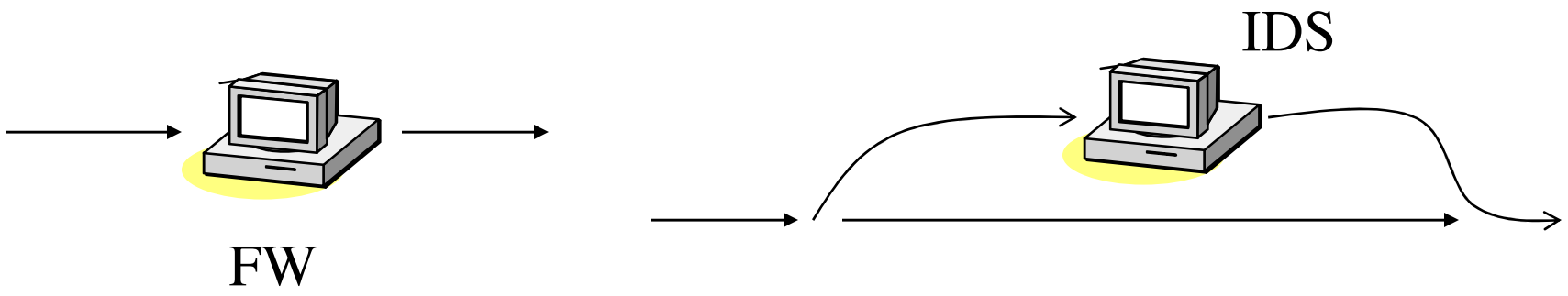
- Deploying sensors at strategic locations
  - E.G., Packet sniffing via *tcpdump* at routers
- Inspecting network traffic
  - Watch for violations of protocols and unusual connection patterns
- Monitoring user activities
  - Look into the data portions of the packets for malicious command sequences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

Architecture of Network IDS



Firewall Versus Network IDS

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



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!

Case Study: Snort IDS

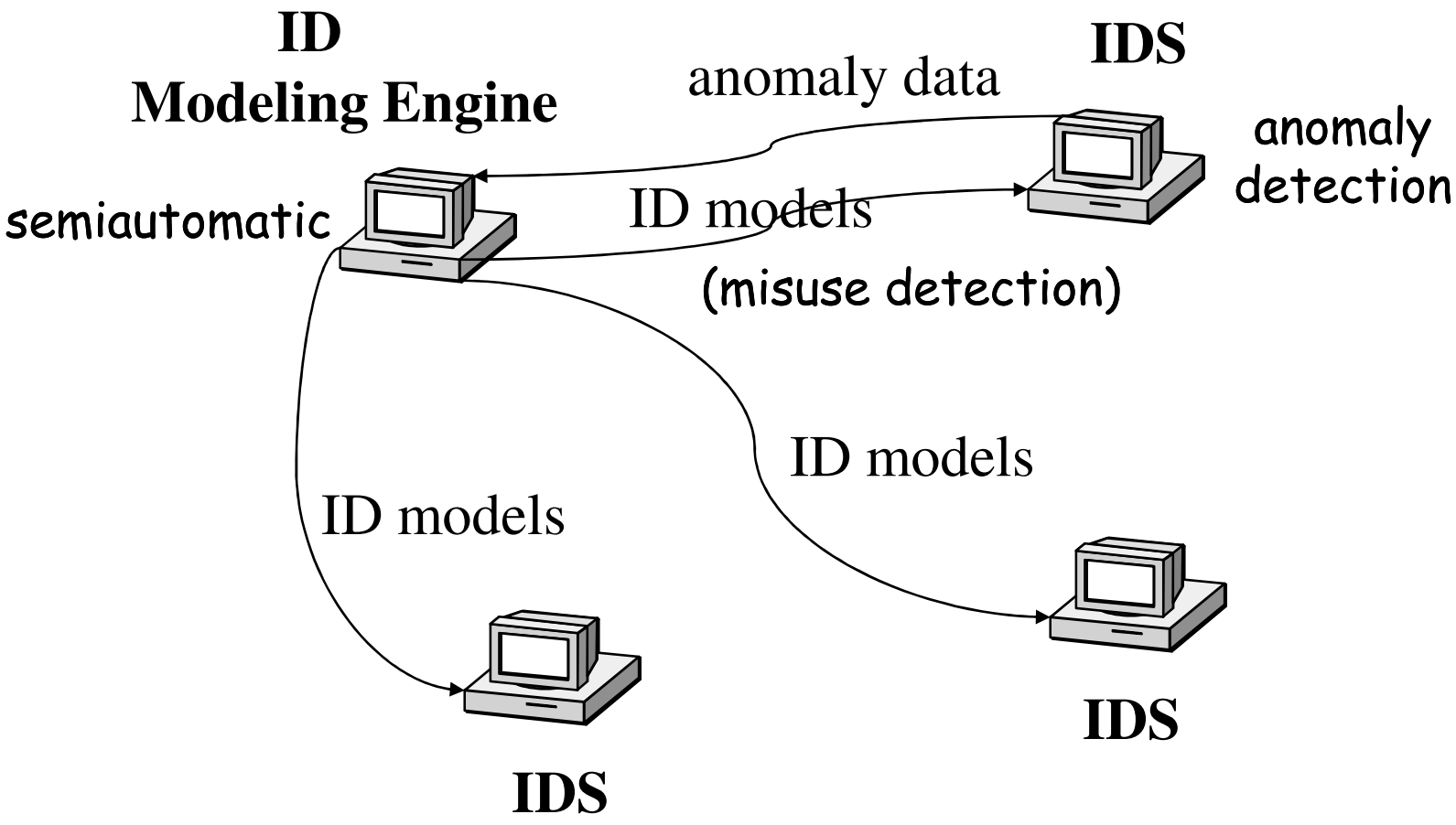
Problems with Current IDSs

- Knowledge and signature-based:
  - "We have the largest knowledge/signature base"
  - Ineffective against new attacks
- Individual attack-based:
  - "Intrusion *A* detected; Intrusion *B* detected …"
  - No long-term proactive detection/prediction
- Statistical accuracy-based:
  - "*x*% detection rate and *y*% false alarm rate"
    - Are the *most damaging* intrusions detected?
- Statically configured.

Next Generation IDSs

- Adaptive
  - Detect new intrusions
- Scenario-based
  - Correlate (multiple sources of) audit data and attack information
- Cost-sensitive
  - Model cost factors related to intrusion detection
  - Dynamically configure IDS components for best protection/cost performance

Adaptive IDSs



Semi-automatic Generation of ID Models