Intrusion Detection
Notes- Set13

Misuse Prevention

- **Prevention techniques**: first line of defense
- Secure local and network resources
- Techniques: cryptography, identification, authentication, authorization, access control, security filters, etc.

Problem: **Losses occur**!

Contributing Factors for Misuse

- Many **security flaws** in systems
- Secure systems are **expensive**
- Secure systems are **not user-friendly**
- “Secure systems” **still have flaws**
- **Insider** Threat
- **Hackers’** skills and tools improve

Intrusion

- An authorized action ...
  - that exploits a vulnerability ...
  - that causes a compromise ...
  - and thus a successful attack.
- *Authentication and Access Control Are No Help!*

Need:

- **Intrusion Prevention**: protect system resources
- **Intrusion Detection**: (second line of defense) discriminate intrusion attempts from normal system usage
- **Intrusion Recovery**: cost effective recovery models

Why Intrusion Detection?

- Second line of defense
- Deter intruders
- Catch intruders
- Prevent threats to occur (real-time IDS)
- Improve prevention/detection techniques

IDS vs Access Control

- IDS rules describe
  - subjects (sources), objects (addresses and ports),operations (send/receive)
    - Like access control
- But, also
  - Argument values
  - Order of messages
  - Protocols
- Claim: IDS is more complex than access control
  - IDS allows access, but tries to determine intent
  - Allow a move in chess, but predict impact

Intrusion Detection - Milestones

- **1980**: Deviation from historical system usage (Anderson)
- **1987**: framework for general-purpose intrusion detection system (Denning)
- **1988**: intrusion detection research splits
  - Attack signatures based detection (MIDAS)
  - Anomaly detection based detection (IDES)

Intrusion Detection - Milestones

- **Early 1990s**: Commercial installations
  - IDES, NIDES (SRI)
  - Haystack, Stalker (Haystack Laboratory Inc.)
  - Distributed Intrusion Detection System (Air Force)
- **Late 1990s - today**:
  - Integration of audit sources
  - Network based intrusion detection
  - Hybrid models
  - Immune system based IDS

Terminology

- **Audit**: activity of looking at user/system behavior, its effects, or the collected data
- **Profiling**: looking at users or systems to determine what they usually do
- **Anomaly**: abnormal behavior
- **Misuse**: activity that violates the security policy
- **Outsider**: someone without access right to the system
- **Insider**: someone with access right to the system
- **Intrusion**: misuse by outsiders and insiders

Phases of Intrusion

- **Intelligence gathering**: attacker observes the system to determine vulnerabilities
- **Planning**: attacker decide what resource to attack (usually least defended component)
- **Attack**: attacker carries out the plan
- **Hiding**: attacker covers tracks of attack
- **Future attacks**: attacker installs backdoors for future entry points

Times of Intrusion Detection

- Real-time intrusion detection
  - Advantages:
    - May detect intrusions in early stages
    - May limit damage
  - Disadvantages:
    - May slow down system performance
    - Trade off between speed of processing and accuracy
    - Hard to detect partial attacks

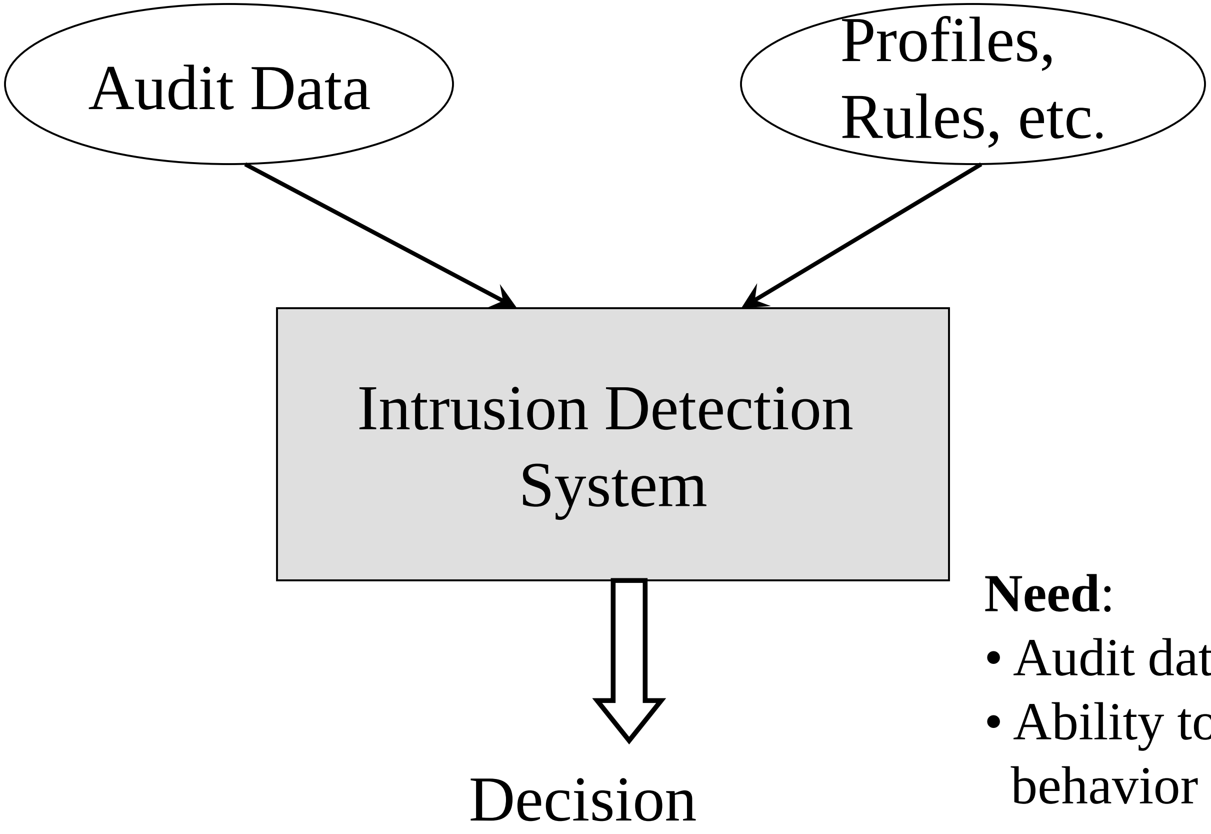
Times of Intrusion Detection

- Off-the-line intrusion detection
  - Advantages:
    - Able to analyze large amount of data
    - Higher accuracy than real-time ID
  - Disadvantages:
    - Mostly detect intrusions after they occurred

Audit Data

- Format, granularity and completeness depend on the collecting tool
- Examples
    - System tools collect data (login, mail)
    - Additional collection of low system level
    - “Sniffers” as network probes
    - Application auditing
- Needed for
    - Establishing guilt of attackers
    - Detecting subversive user activity

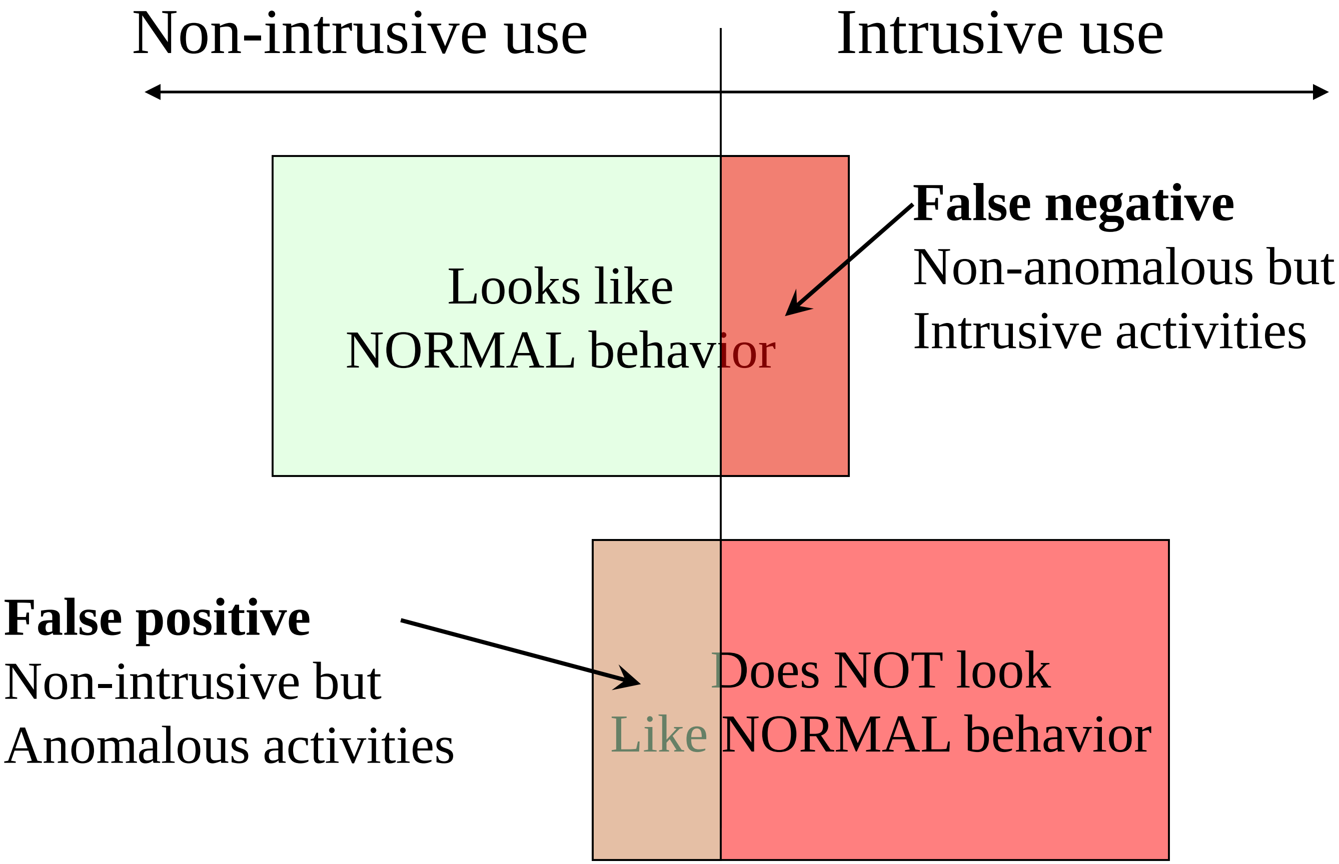
Audit-Based Intrusion Detection



Need:

- Audit data
- Ability to characterize behavior

Anomaly versus Misuse



Non-intrusive use

Intrusive use

Looks like NORMAL behavior

False negative
Non-anomalous but
Intrusive activities

False positive
Non-intrusive but
Anomalous activities

Does NOT look Like NORMAL behavior

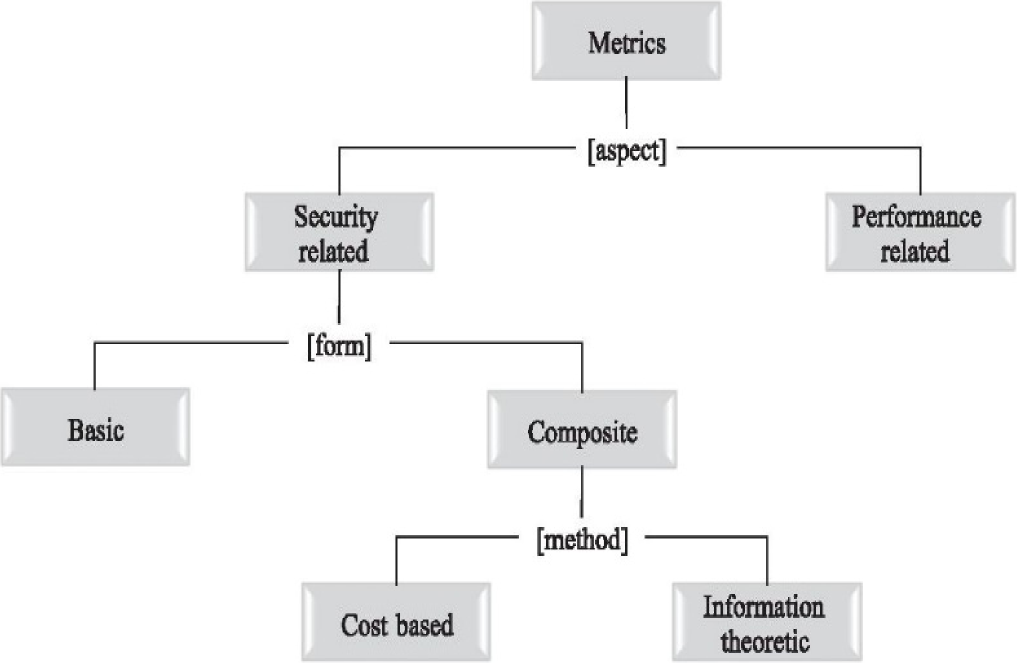
ERRORS

- **A False Positive error occurs when a non-intrusion causes an alarm**
- **A False Negative error occurs when an intrusion does not result in an alarm.**
- **False Positive Rate (FPR) = #False Positives / #Normal Events**
- **False Negative Rate (FNR) = #False Negatives / #Intrusions**

False Positive v.s. False Negative

- **False positive**: non-intrusive but anomalous activity
  - Security policy is not violated
  - Cause unnecessary interruption
  - May cause users to become unsatisfied
- **False** negative: non-anomalous but intrusive activity
  - Security policy is violated
  - Undetected intrusion

Metrics


Metrics
[aspect]
Security related
Performance related
[form]
Basic
Composite
[method]
Cost based
Information theoretic

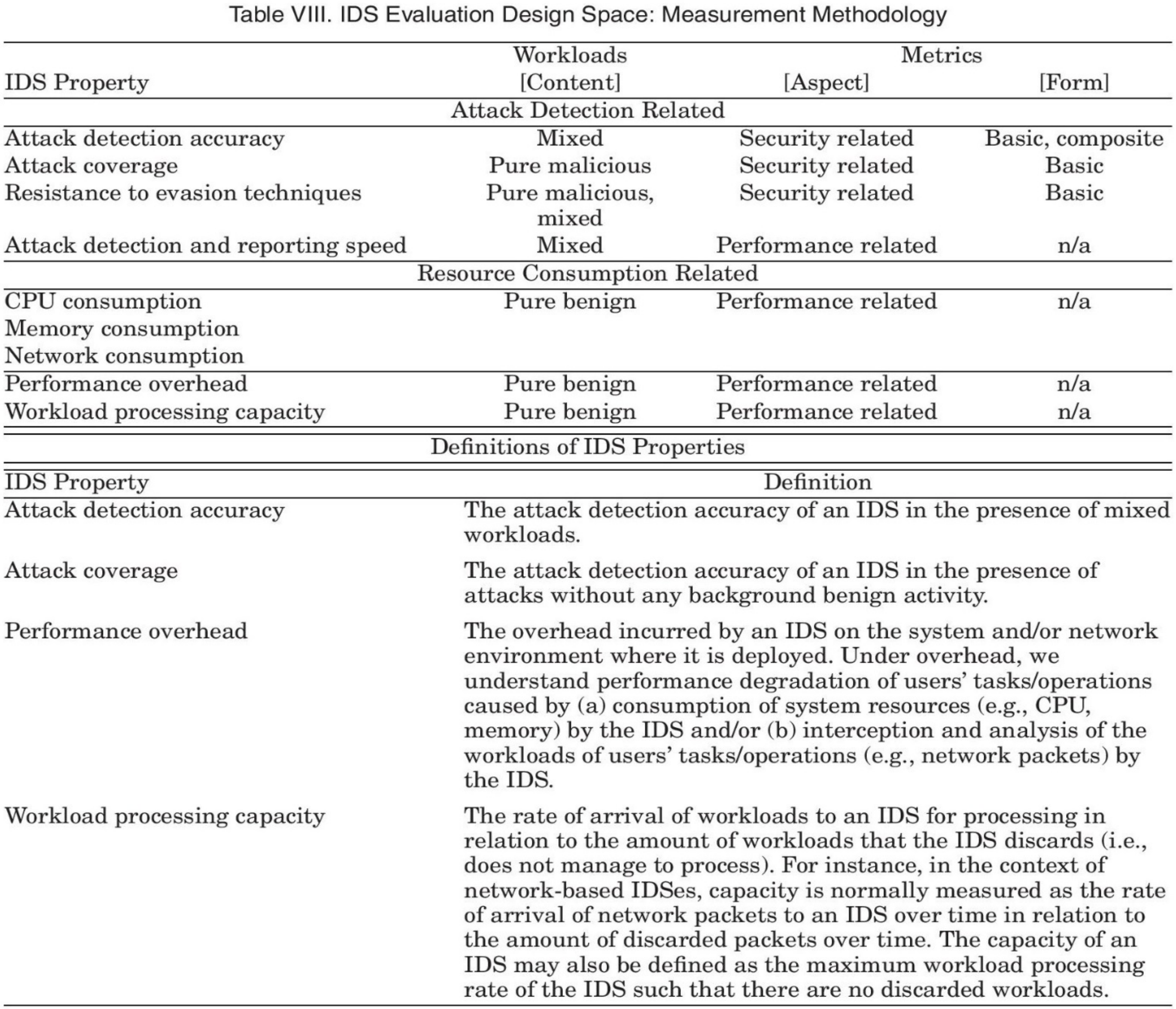


Table VIII. IDS Evaluation Design Space: Measurement Methodology

	Workloads	Metrics	
IDS Property	[Content]	[Aspect]	[Form]
	Attack Detection Related		
Attack detection accuracy	Mixed	Security related	Basic, composite
Attack coverage	Pure malicious	Security related	Basic
Resistance to evasion techniques	Pure malicious, mixed	Security related	Basic
Attack detection and reporting speed	Mixed	Performance related	n/a
	Resource Consumption Related		
CPU consumption Memory consumption Network consumption	Pure benign	Performance related	n/a
Performance overhead	Pure benign	Performance related	n/a
Workload processing capacity	Pure benign	Performance related	n/a

Definitions of IDS Properties

IDS Property	Definition
Attack detection accuracy	The attack detection accuracy of an IDS in the presence of mixed workloads.
Attack coverage	The attack detection accuracy of an IDS in the presence of attacks without any background benign activity.
Performance overhead	The overhead incurred by an IDS on the system and/or network environment where it is deployed. Under overhead, we understand performance degradation of users' tasks/operations caused by (a) consumption of system resources (e.g., CPU, memory) by the IDS and/or (b) interception and analysis of the workloads of users' tasks/operations (e.g., network packets) by the IDS.
Workload processing capacity	The rate of arrival of workloads to an IDS for processing in relation to the amount of workloads that the IDS discards (i.e., does not manage to process). For instance, in the context of network-based IDSes, capacity is normally measured as the rate of arrival of network packets to an IDS over time in relation to the amount of discarded packets over time. The capacity of an IDS may also be defined as the maximum workload processing rate of the IDS such that there are no discarded workloads.

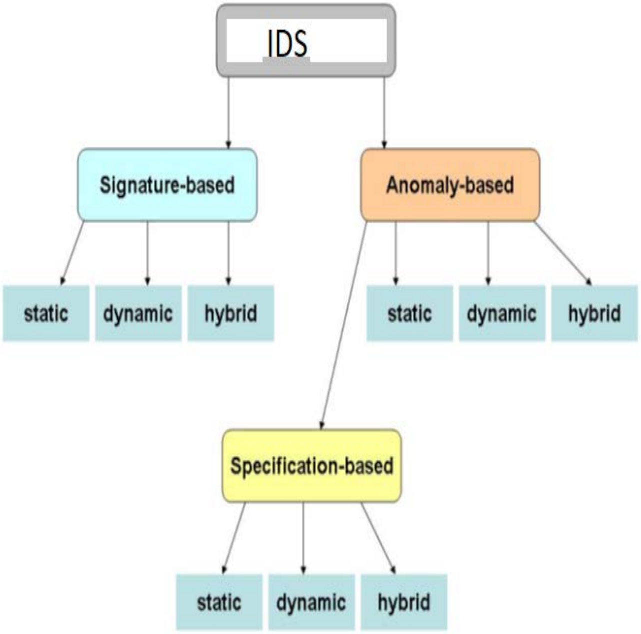
Milenkoski, A., Vieira, M., Kounev, S., Avritzer, A. and Payne, B.D., 2015. Evaluating computer intrusion detection systems: A survey of common practices. ACM Computing Surveys (CSUR), 48(1)

Intrusion Detection Techniques

1. Anomaly Detection
2. Misuse Detection
3. Hybrid Misuse/Anomaly Detection
4. Immune System Based IDS

Intrusion Detection Techniques

I know what is bad and can detect it
False positives: none
False negatives: ever increasing



I will learn what is good and bad
False positives: incorrect learning
False negatives: incorrect learning

I know what is good and can detect when you go beyond specification
False positives: incomplete specification
False negatives: incorrect specification

Nwokedi Idika and Aditya Mathur, A Survey of Malware Detection Techniques, Purdue University, Feb 2007.

Rules and Profiles

- **Statistical techniques**:
  - Collect usage data to statistically analyze data
  - Good for both anomaly-based and misuse-based detection:
    - Anomaly-based: standards for normal behavior. Warning when deviation is detected
    - Misuse-based: standards for misuse. Warning when phases of an identified attack are detected
  - Threshold detection
    - E.g., number of failed logins, number of accesses to resources, size of downloaded files, etc.

Rules and Profiles

- Rule-based techniques:
  - Define rules to describe normal behavior or known attacks
  - Good for both anomaly-based and misuse-based detection:
    - Anomaly-based: looks for deviations from previous usage
    - Misuse-based: define rules to represent known attacks

Misuse Detection

- Profile signatures of known attacks
  - Monitor operational state for signature
  - Hypothesis: attacks of the same kind has enough similarity to distinguish from normal behavior
  - This is largely *pattern matching*
- Q: Where do these signatures come from?
  - Record: recorded progression of known attacks
  - Expert: domain knowledge
- AI: Learn by negative and positive feedback

Misuse Detection Techniques

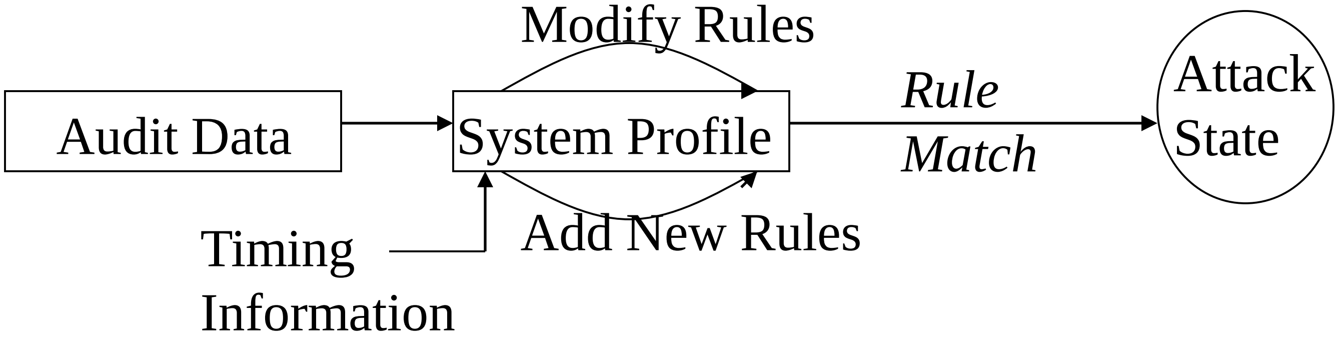
Represent attacks in the form of pattern or a signature (variations of same attack can be detected)

Problem!

Cannot represent new attacks

Misuse Detection Techniques

- Expert Systems
- Model Bases Reasoning
- State Transition Analysis
- Neutral Networks

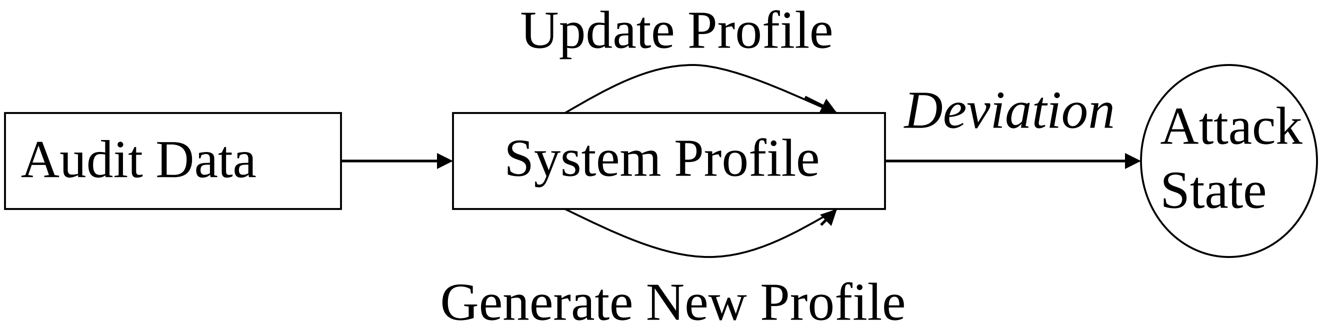


SIGNATURE MATCHING

- **Advantages:**
  - **very low false positive rate**
  - **Automated extraction is possible**
- **Disadvantages**
  - **Only detects already known attacks**
  - **Simple changes to an attack can defeat detection, for example:**
  - **scan even ports, then odd ports.**
  - **“rm –rf /*” -> “rm –rf ../../../../../../../../*”**

Anomaly Detection Techniques

- Assume that all intrusive activities are necessarily anomalous → flag all system states that very from a “normal activity profile”
- Need:
  - Selection of features to monitor
  - Good threshold levels to prevent false-positives and false-negatives
  - Efficient method for keeping track and updating system profile metrics



ANOMALY DETECTION

- **1.Try to identify what is normal, e.g. “normal” command sequence or common 4-tuple/protocol, session length, intervals, etc…**
- **2. Look for major deviations (outliers), e.g. unusual target port, source addr, or port sequence (scan)**
- **Sometimes (but not always): Apply AI/Machine Learning to “learn” what is normal.**
- **Advantage: more robust to “altered” attacks.**

Anomaly Detection

- Compares profile of normal systems operation to monitored state
  - Hypothesis: any attack causes enough deviation from profile (generally true?)
- Q: How do you derive normal operation?
  - AI: learn operational behavior from training data
  - Expert: construct profile from domain knowledge
  - Black-box analysis (vs. white or grey?)
- Q: Is normal the same for all environments?
- Pitfall: *false learning*

ANOMALOUS PROBLEMS

- **High false positive rate**
- **Attacks might not be obvious until too late**
- **Attacks can hide in “normal” traffic**
- **Require training on “known good” data**
- **Problems when what’s “normal” changes**
    - **Eg, flash crowds, new users, new applications…**

HOST-BASED IDS

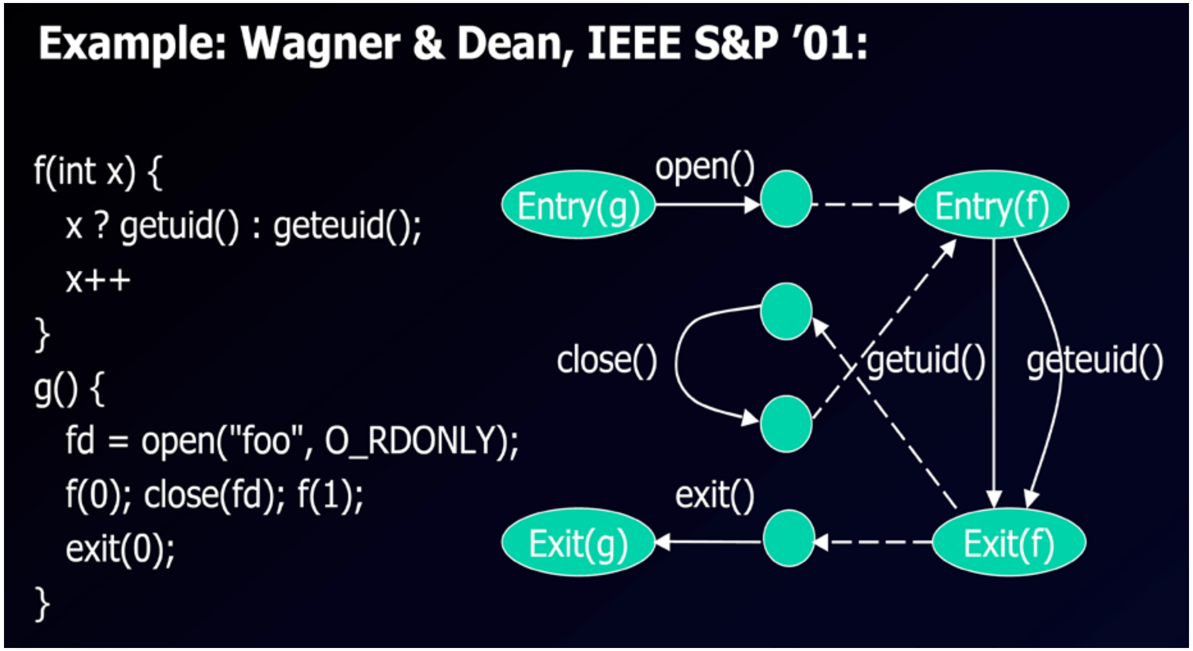
- **… monitor a single host to detect intrusions.**
- **Typical “sensors” include:**
  - **– Disk & memory contents**
  - **– User input and commands**
  - **– System calls**
- **… Typically equate “intrusion” with privilege escalation, eg remote ! local ! root.**
- **… Are vulnerable to various attacks, e.g.: replace HIDS process/binaries, “Mimic” attack, context problems, subvirtion.**

EXAMPLE: TRIPWIRE

- **A “standard attack” might go like this:**
  - **Gain user access to system**
  - **Gain root access**
  - **Replace system binaries to set up backdoor**
  - **Use backdoor for future activities**
- **Tripwire is a simple anomaly-based HIDS that monitors system binaries:**
  - **Compute hash of key system binaries**
  - **Compare current hash to earlier stored hash**
  - **Report problem if hash is different**
  - **Store reference hash codes on read-only medium**
- **Example attack: “rootkit” replaces binaries with new versions with the same hash.**

APPLICATION-BASED IDS

- **Idea: Build NFA for each application, from source code.**



Example: Wagner & Dean, IEEE S&P ’01:

```
f(int x) {
   x ? getuid() : geteuid();
   x++
}
g() {
   fd = open("foo", O_RDONLY);
   f(0); close(fd); f(1);
   exit(0);
}
```

NETWORK IDS

- **A network-based IDS monitors network traffic for intrusions, e.g.:**
  - **DoS**
  - **Exploiting bugs in protocol, application, OS**
  - **Install worm, virus, bot, spyware…**
- **Challenges for NIDS: fragmentation, data volume, “low and slow” attacks…**

Example

- **A typical attack is “port scanning” to find out what services a host is running.**
- **For exampe, the Nmap tool: (http://www.insecure.org/nmap/)**
    - **Determines OS/hostname/device type via “service fingerprinting” (eg, IRIX listens on TCP port 1)**
    - **Determines what service is listening on a port and can determine application name, version**
    - **Operates in optional obfuscation mode**
- **Afterwards, attackers can exploit known vulnerabilities in the OS/applications found via Nmap.**

EXAMPLE: SNORT

- Snort - Open source IDS
- Started by Martin Roesch in 1998 as a lightweight IDS
- Snort rules
  - Sample: alert tcp any any -> 192.168.1.0/24 111 (content:"|00 01 86 a5|"; msg: "mountd access";)
  - Rule Header: Action, Protocol, Src+Port -> Dest+Port
  - Rule Options: Alert messages and Packet Content
- **… is a signature-based, portable, opensource NIDS with over 2 million downloads, and 100K active users.**
- **… scans a tcpdump log, and finds connections matching attack signatures.**
- **… uses regular expressions that match known attacks.**
- **Example sig:**
  - **alert tcp any any -> [a.b.0.0/16,c.d.e.0/24] 80**

    (msg:"WEB-ATTACKS conf/httpd.conf attempt"; nocase; sid:1373; flow:to_server,established; content:"conf/httpd.conf"; [...])

EXAMPLE: BRO

- **… is a high-speed "policy-based" NIDS. It uses scripts that monitor connections, and check for conformance to "expected" behavior based on protocol**
- **… logs, for all TCP connections (via SYN/FIN/RST packets): start time, duration, service, addresses, sizes, etc. It also identifies the application based on these packets.**
- **… supports many standard applications: DNS, FTP, HTTP, SMTP, NTP, …**
- **… on Telnet and Rlogin generates the following events:**
  - **login_successful, login_failure, activating_encryption, login_confused, login_input_line, login_output_line**
- **In its first five months of operation, Bro found 120 UCB break-ins (60 root compromises)**

Hybrid Misuse / Anomaly Detection

- Anomaly and misuse detection approaches together
- Example:
  1. Browsing using “nuclear” is not misuse but might be anomalous
  2. Administrator accessing sensitive files is not anomalous but might be misuse

Immune System Based ID

- Detect intrusions by identifying suspicious changes in system-wide activities.
- System health factors:
  - Performance
  - Use of system resources
- Need: identify system-wide measurements

Immune System Based ID

- Principal features of human immune system that are relevant to construct robust computer systems:
  1. Multi-layered protection
  2. Distributed detection
  3. Diversity of detection
  4. Inexact matching ability
  5. Detection of unseen attacks

DISTRIBUTED IDS

- **Idea: combine data from many sensors**
  - **HIDS data from many hosts**
  - **NIDS data from multiple segments**
- **Positives: can detect “stealthy” scans, possibly lower false alarm rate, etc.**
- **Negatives: much more data to scan, exchange, etc.**

HONEYPOTS & TARPITS

- **A honeypot is a closely monitored network decoy or decoy(s). Honeypots:**
  - **Distract adversaries from more valuable machines on a network (?)**
  - **Provide early warning about new attack and exploitation trends**
  - **Frequently consist of multiple VMs that use up the “empty” IP address space on a network.**
- **A tarpit is essentially a slow honeypot that accepts connections, but prevents resets.**

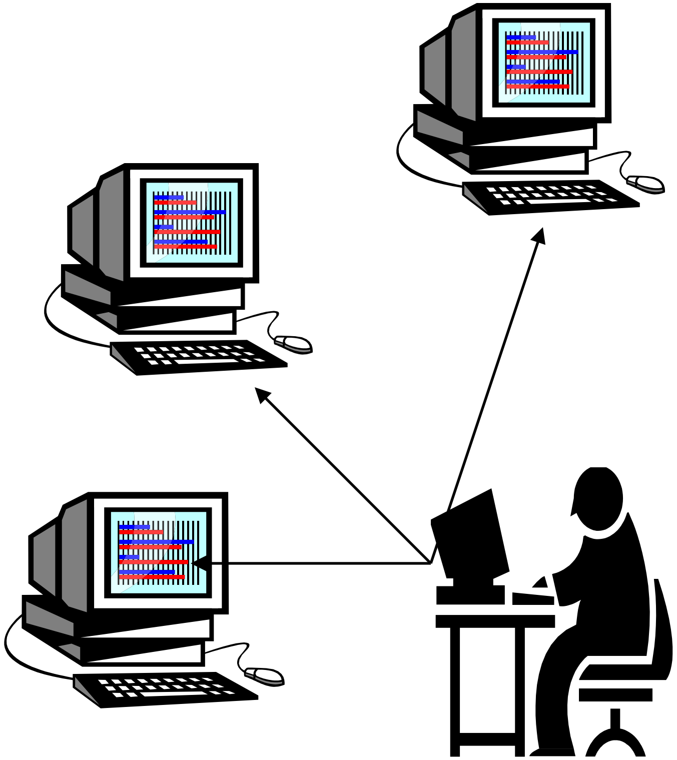
Intrusion Types

- Doorknob rattling
- Masquerade attacks
- Diversionary Attack
- Coordinated attacks
- Chaining
- Loop-back

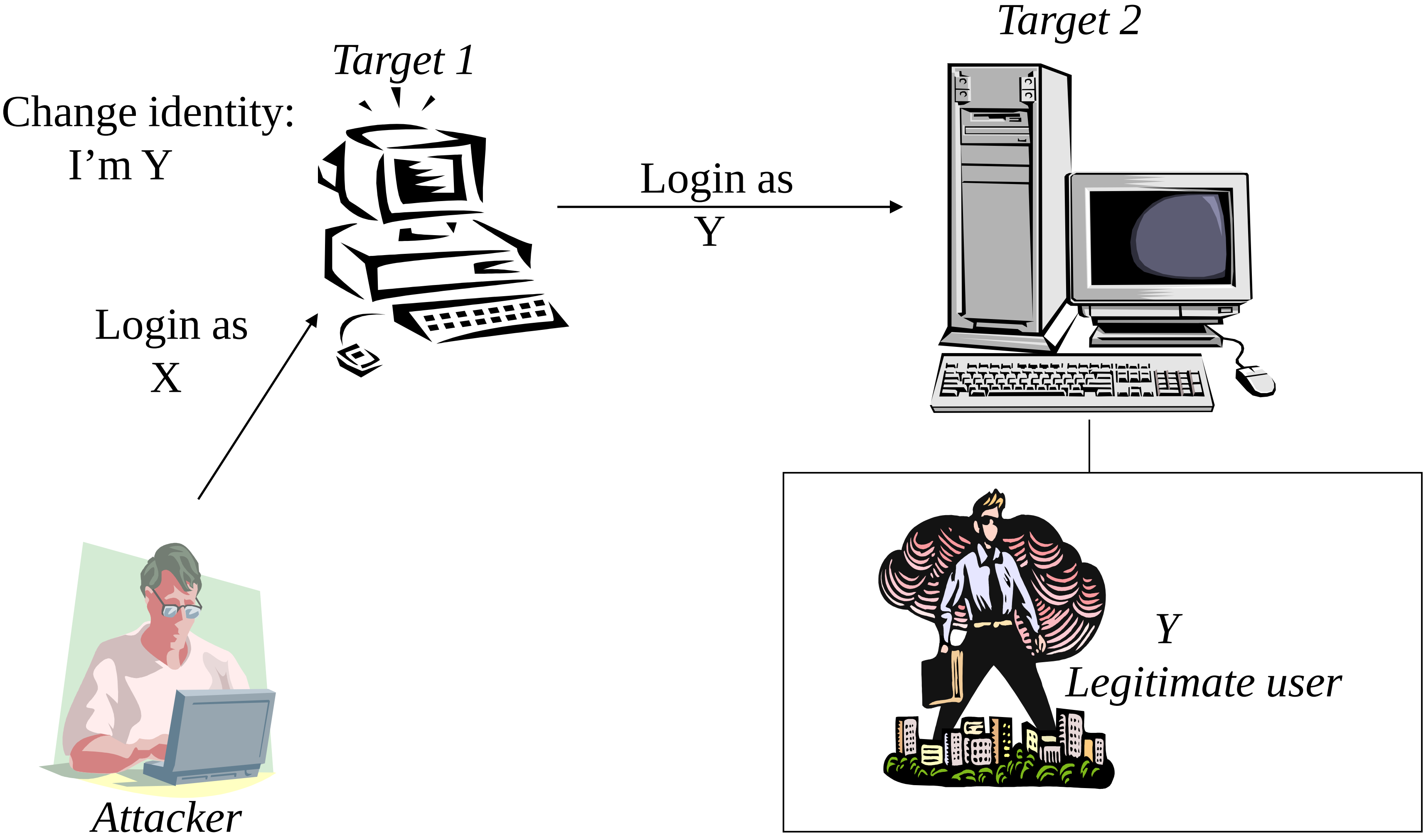
Doorknob Rattling

- Attack on activity that can be audited by the system (e.g., password guessing)
- Number of attempts is lower than threshold
- Attacks continue until
  - All targets are covered

  or
  - Access is gained



Masquerading



Target 1

Change identity:
I’m Y

Target 2

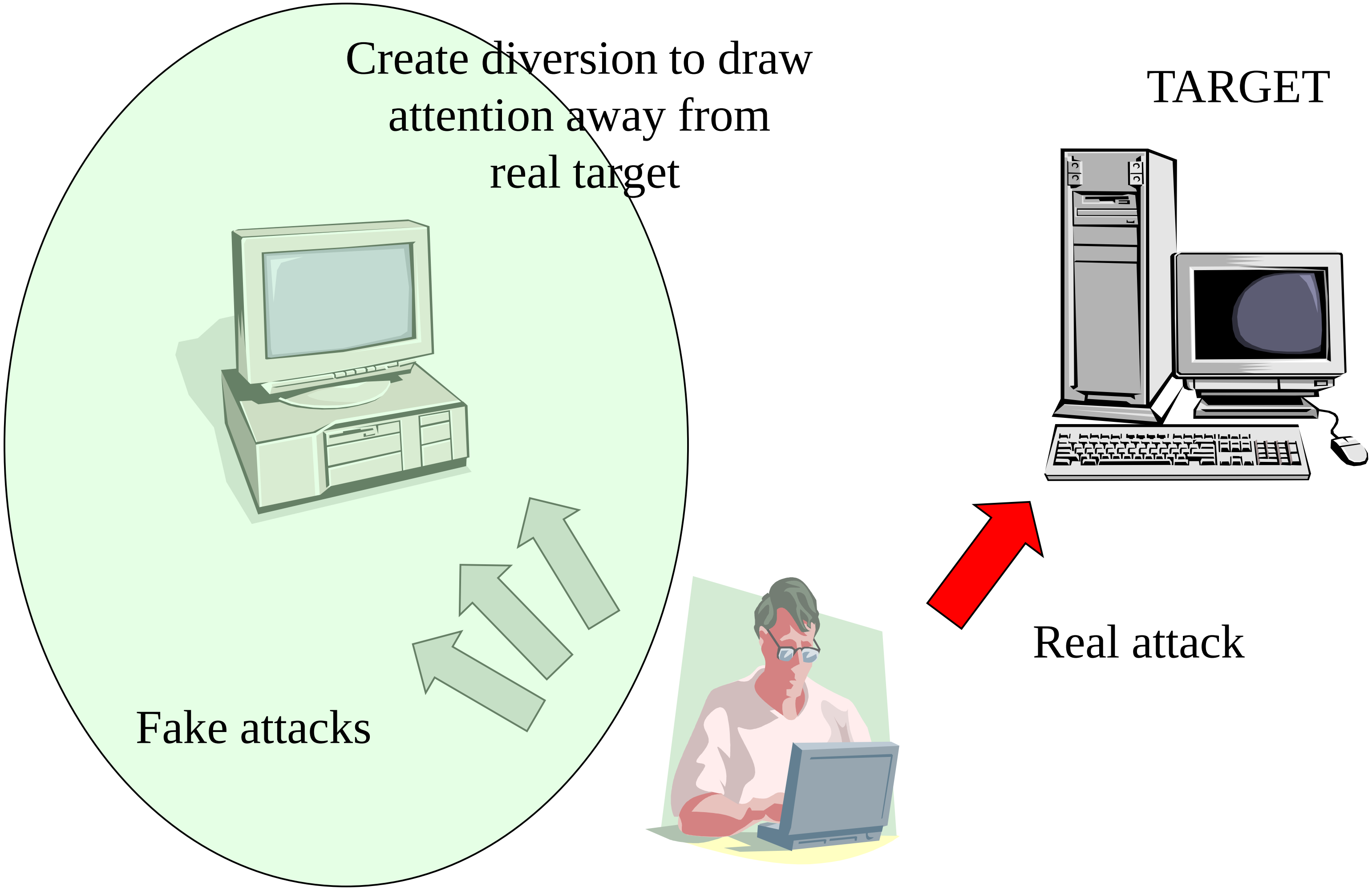
Login as
Y

Login as
X

Y
Legitimate user

Attacker

Diversionary Attack



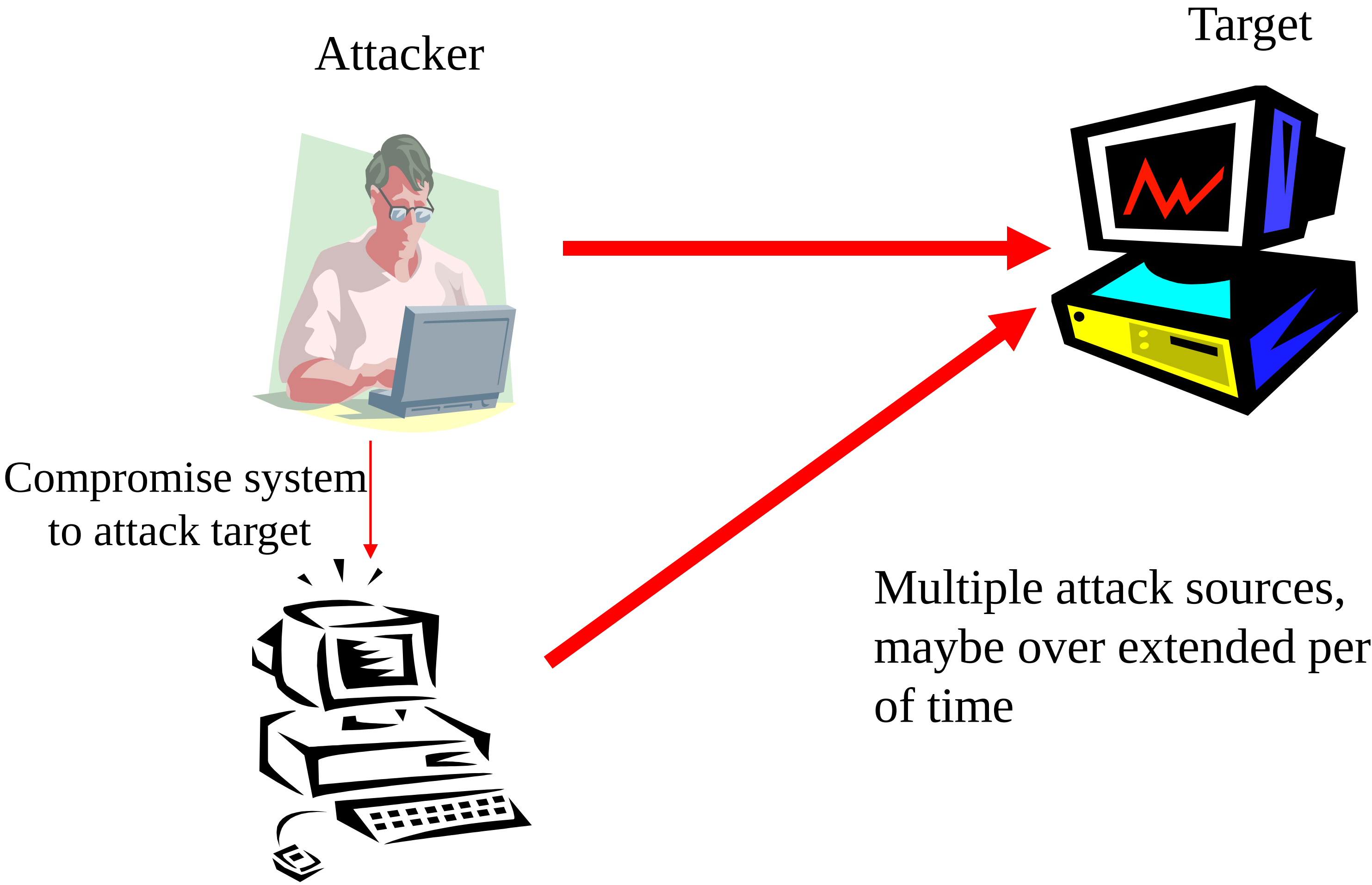
Create diversion to draw attention away from real target

TARGET

Fake attacks

Real attack

Coordinated attacks



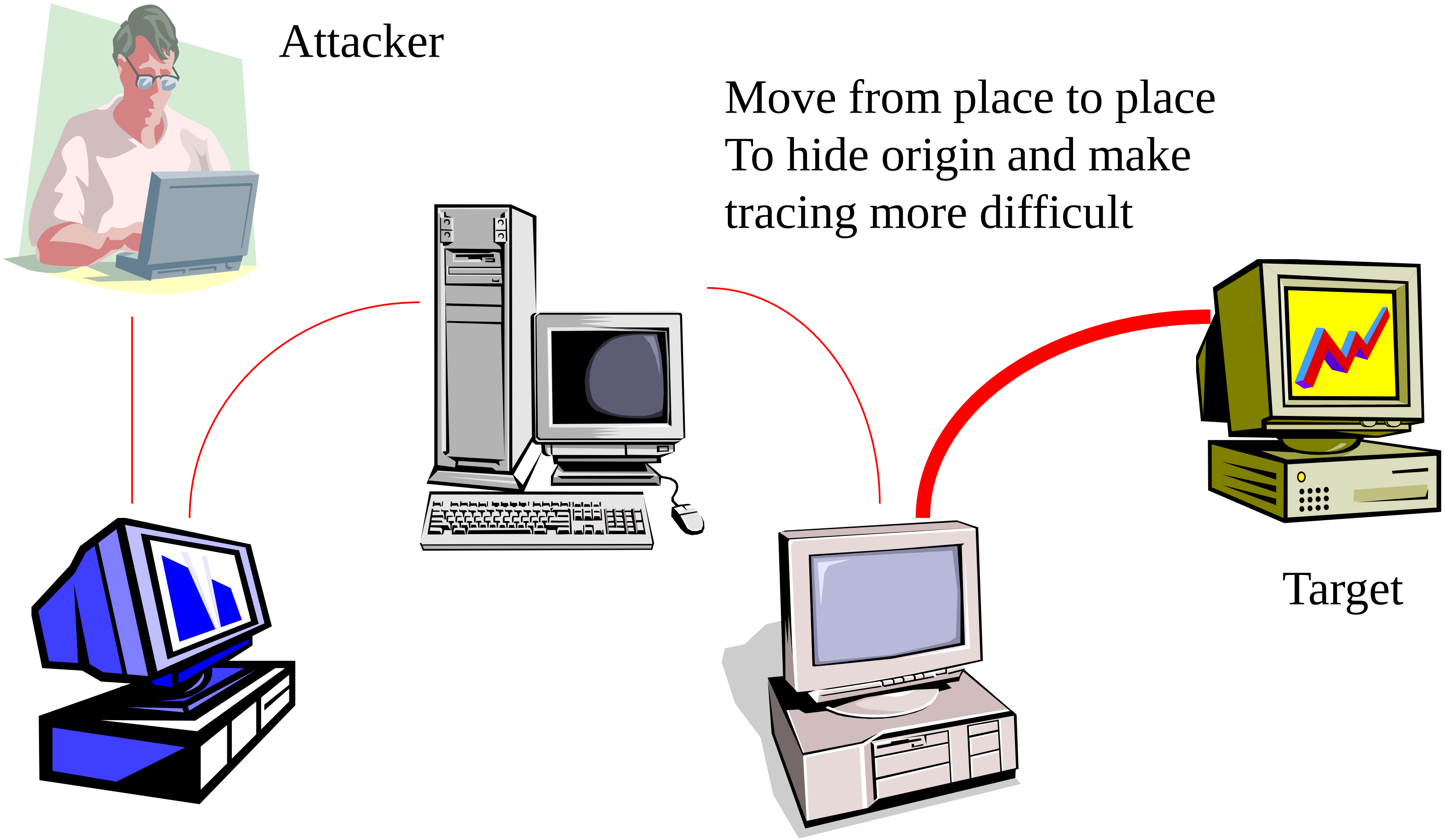
Attacker

Target

Compromise system to attack target

Multiple attack sources, maybe over extended period of time

Chaining



Attacker

Move from place to place
To hide origin and make
tracing more difficult

Target

Intrusion Recovery

- Actions to avoid further loss from intrusion.
- Terminate intrusion and protect against reoccurrence.
- Reconstructive methods based on:
    - Time period of intrusion
    - Changes made by legitimate users during the effected period
    - Regular backups, audit trail based detection of effected components, semantic based recovery, minimal roll-back for recovery.

Next Class

- Economic and legal aspects