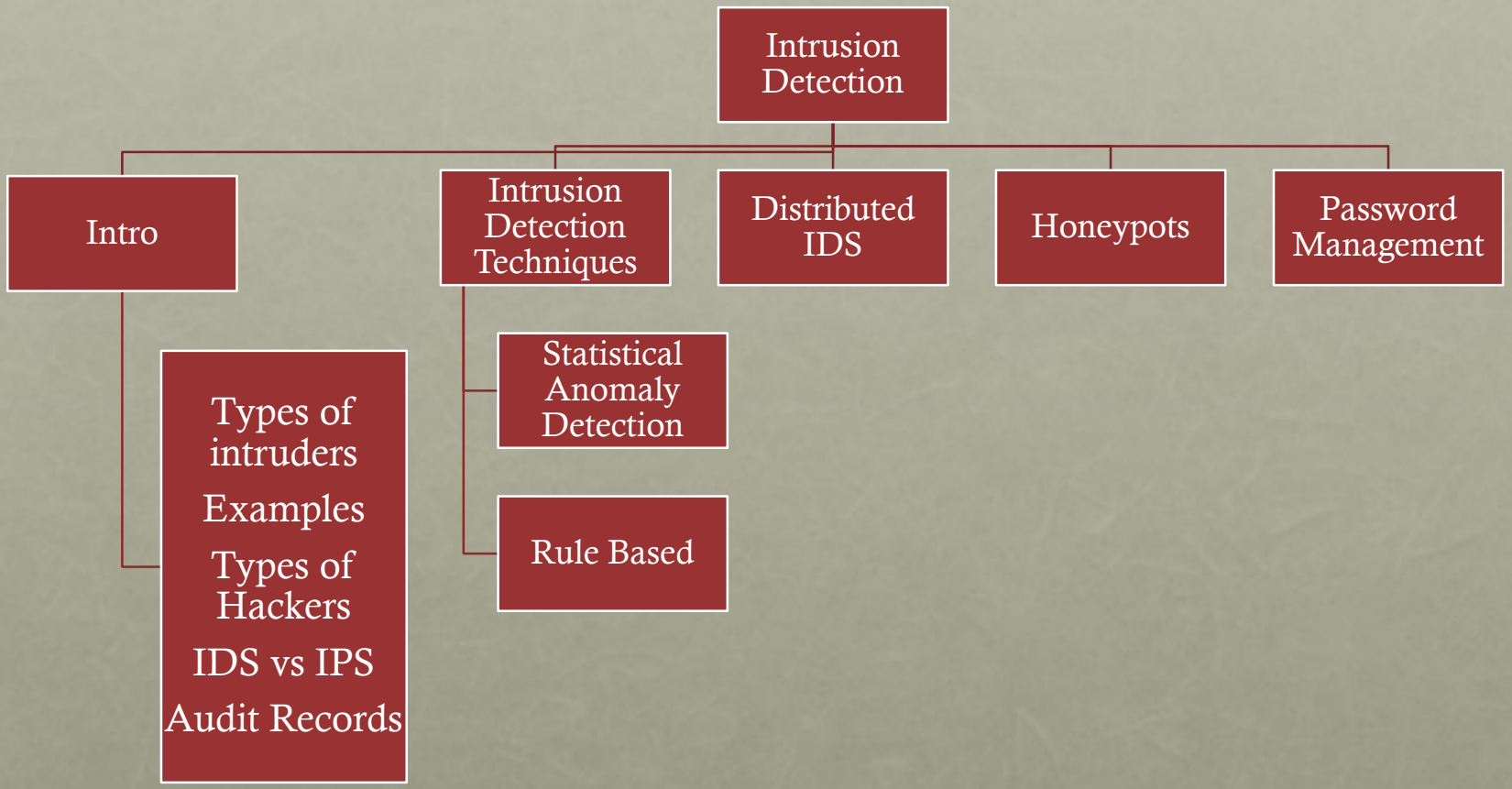
CHAPTER 11

Intrusion detection systems

OUTLINE


Intrusion Detection
Intro
Types of intruders
Examples
Types of Hackers
IDS vs IPS
Audit Records
Intrusion Detection Techniques
Statistical Anomaly Detection
Rule Based
Distributed IDS
Honeypots
Password Management

INTRUDERS

- Three classes of intruders:

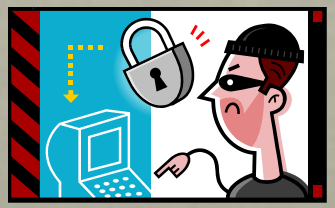
Masquerader	• An individual who is not authorized to use the computer and who penetrates a system's access controls to exploit a legitimate user's account
Misfeasor	• A legitimate user who accesses data, programs, or resources for which such access is not authorized, or who is authorized for such access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

EXAMPLES OF INTRUSION

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

INSIDER ATTACKS

- Among the most difficult to detect and prevent
- Can be motivated by revenge or simply a feeling of entitlement
- Countermeasures:



Enforce least privilege, only allowing access to the resources employees need to do their job

Set logs to see what users access and what commands they are entering

Protect sensitive resources with strong authentication

Upon termination, delete employee's computer and network access

Upon termination, make a mirror image of employee's hard drive before reissuing it (used as evidence if your company information turns up at a competitor

INTRUSION TECHNIQUES

- Objective of the intruder is to gain access to a system or to increase the range of privileges accessible on a system
- Most initial attacks use system or software vulnerabilities that allow a user to execute code that opens a backdoor into the system
- Ways to protect a password file:

One-way functioning	Access control
• The system stores only the value of a function based on the user's password	• Access to the password file is limited to one or a very few accounts

HACKERS

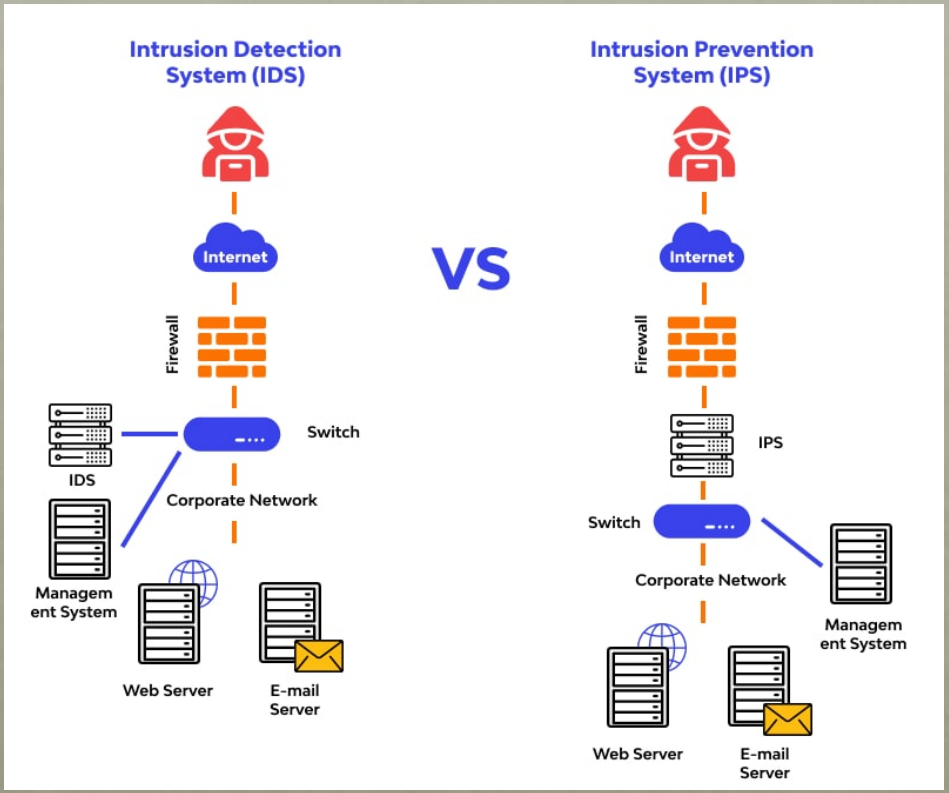
- Organized groups of hackers
- Usually have specific targets, or at least classes of targets in mind
- Once a site is penetrated, the attacker acts quickly, scooping up as much valuable information as possible and exiting
- Can be benign (consume resources) or malignant (damages sensitive data)

CERTS & RED TEAM

- **Intrusion detection systems (IDSs) and intrusion prevention systems (IPSs) are designed to counter hacker threats**
  - **In addition to using such systems, organizations can consider restricting remote logons to specific IP addresses and/or use virtual private network technology**
- **CERTs (Computer emergency response teams)**
  - **These cooperative ventures collect information about system vulnerabilities and disseminate it to systems managers**
  - **Hackers also routinely read CERT reports**
  - **It is important for system administrators to quickly insert all software patches to discovered vulnerabilities**

IDS & IPS

- Intrusion detection systems (IDSs) and intrusion prevention systems (IPSs) are designed to counter hacker threats



- In addition to using such systems, organizations can consider restricting remote logons to specific IP addresses and/or use virtual private network technology

INTRUSION DETECTION

- A system's second line of defense
- Is based on the assumption that the behavior of the intruder differs from that of a legitimate user in ways can be quantified

- Considerations:
  - If an intrusion is detected quickly enough, the intruder can be identified and ejected from the system before any damage is done or any data are compromised
  - Intrusion detection enables the collection of information about intrusion techniques that can be used to strengthen the intrusion prevention facility

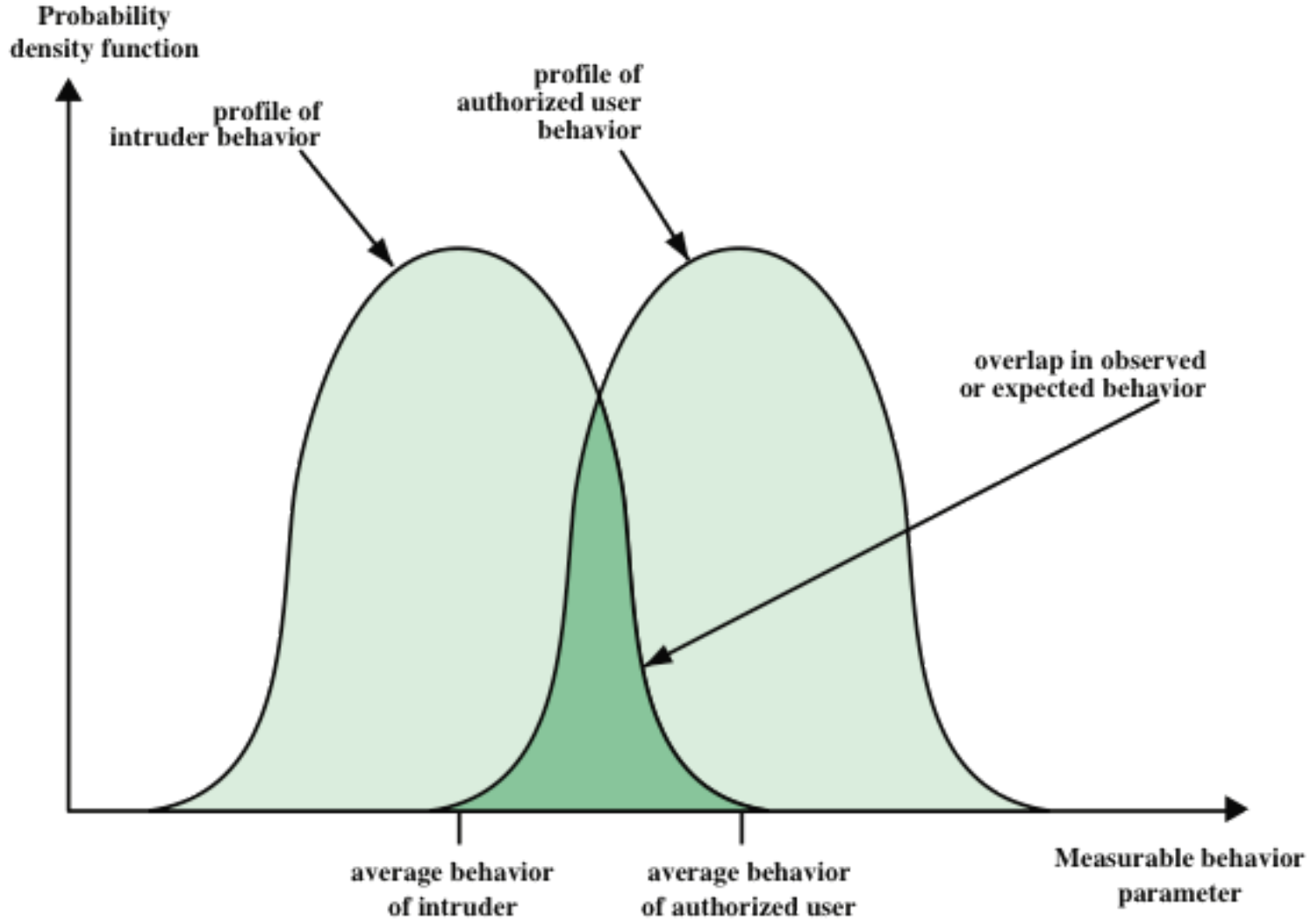


Figure 11.1 Profiles of Behavior of Intruders and Authorized Users

BASE-RATE FALLACY

- To be of practical use, an intrusion detection system 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
  - If the system frequently triggers an alert when there is no intrusion, then either system managers will begin to ignore the alarms or much time will be wasted analyzing the false alarms
- 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
  - 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

AUDIT RECORDS

- Fundamental tool for intrusion detection

Native audit records

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

Detection-specific audit records

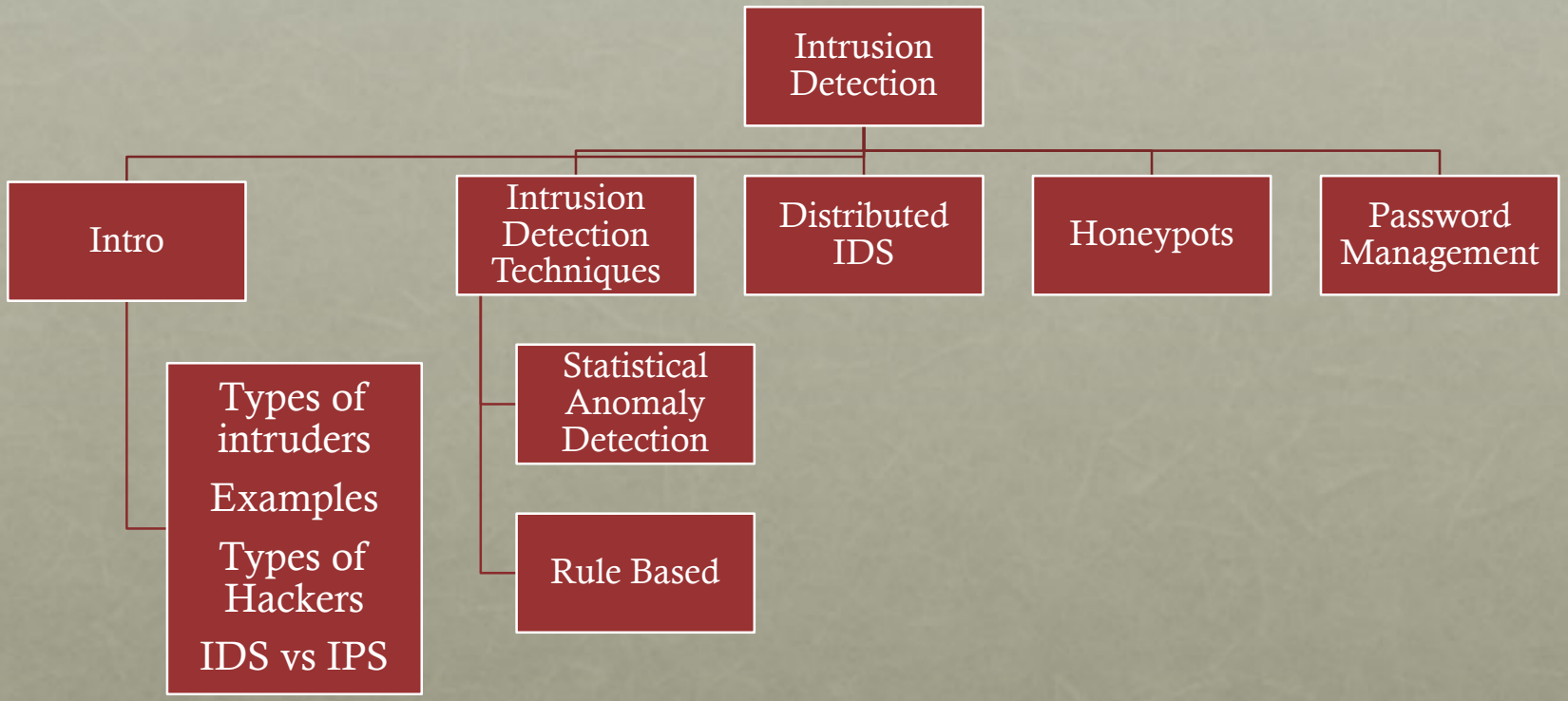
A collection facility can be implemented that generates audit records containing only that information required by the intrusion detection system.

Examples include: Subject, action, object, resource-usage, time-stamp.

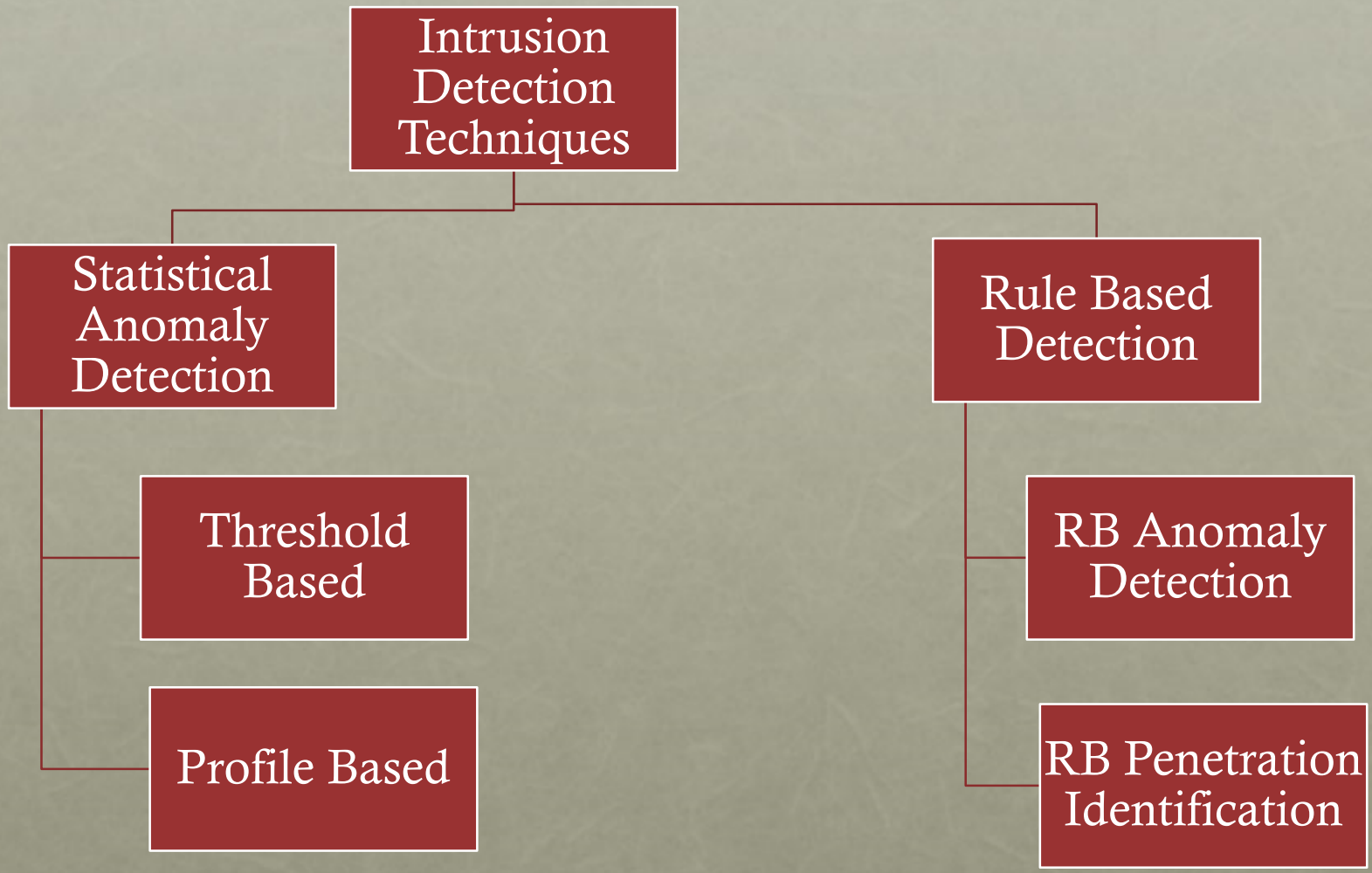
One advantage of such an approach is that it could be made vendor independent and ported to a variety of systems

The disadvantage is the extra overhead involved in having two accounting packages running on a machine

OUTLINE


Intrusion Detection
Intro
Types of intruders
Examples
Types of Hackers
IDS vs IPS
Intrusion Detection Techniques
Statistical Anomaly Detection
Rule Based
Distributed IDS
Honeypots
Password Management

Intrusion Detection Techniques



APPROACHES TO INTRUSION DETECTION

- **Statistical anomaly detection: Effective against Masquerader**
  - **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 whether that behavior is not legitimate user behavior**
  - **Threshold detection**
    - **This approach involves defining thresholds, independent of user, for the frequency of occurrence of various events**
  - **Profile based**
    - **A profile of the activity of each user is developed and used to detect changes in the behavior of individual accounts**
- **Rule-based detection: Effective against Misfeasor**
  - **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**
  - **Often referred to as *signature detection***

STATISTICAL ANOMALY DETECTION

- **Threshold detection**
  - **Involves 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**
  - **By itself is a crude and ineffective detector of even moderately sophisticated attacks**
- **Profile-based**
  - **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**

RULE-BASED INTRUSION DETECTION

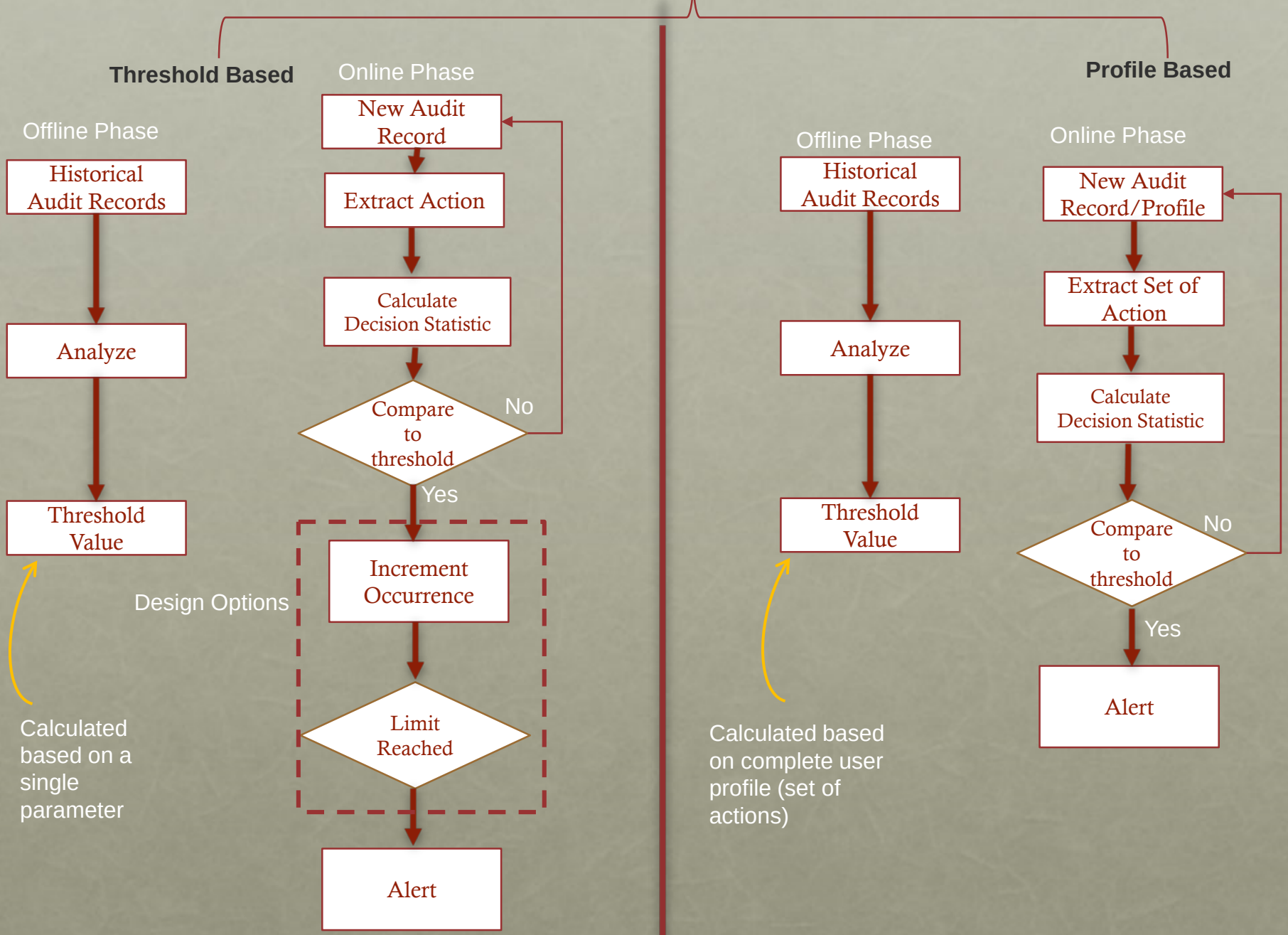
- Techniques detect intrusion by **observing events** in the system and applying a set of rules that lead to a decision regarding whether a given pattern of activity is or is not suspicious

- **Rule-based anomaly detection**
  - Is similar in terms of its approach and strengths to statistical anomaly detection
  - Historical audit records are analyzed to identify usage patterns and to automatically generate rules that describe those patterns
  - 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
  - In order for this approach to be effective, a rather large database of rules will be needed

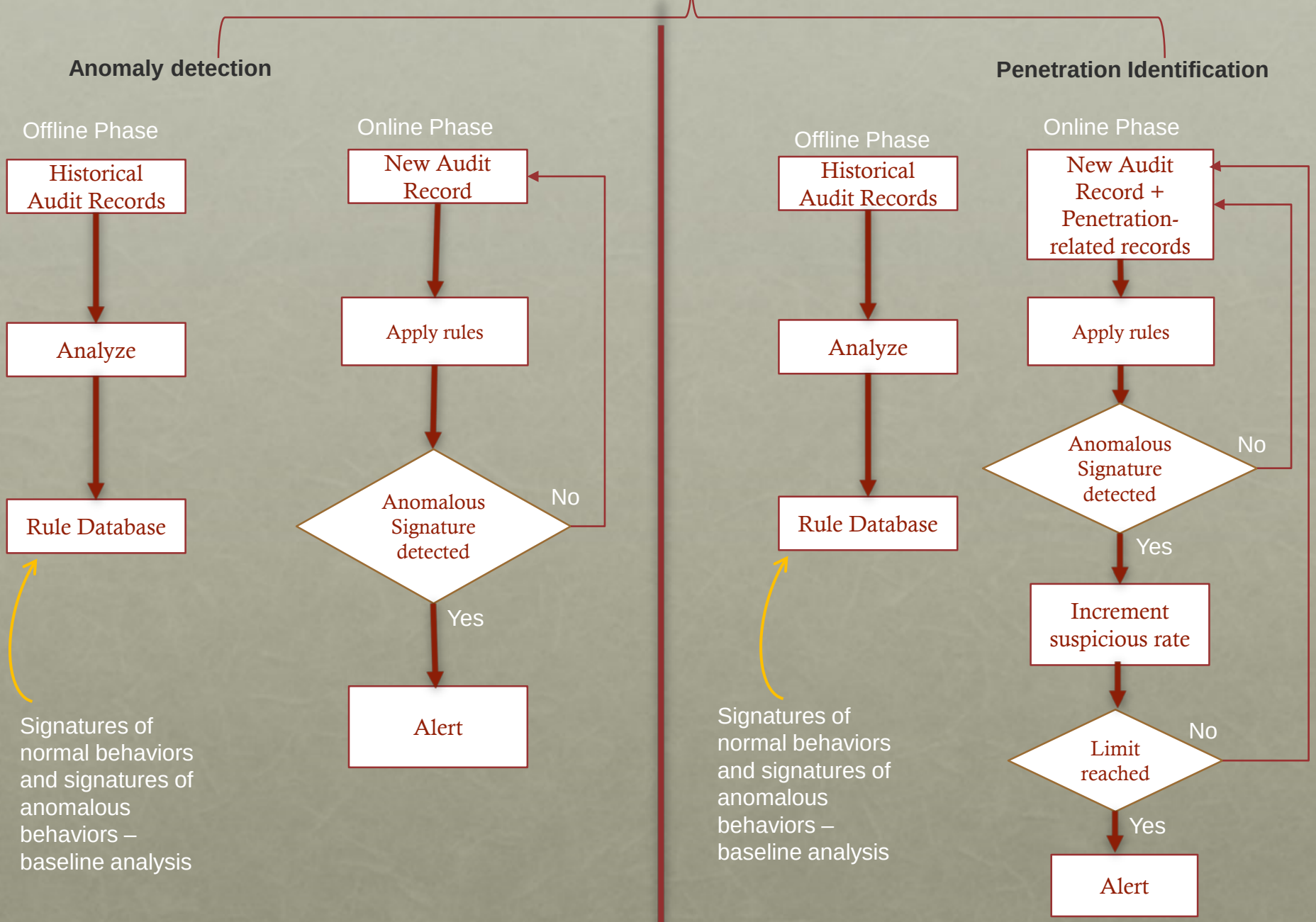
Rule-Based Intrusion Detection

- **Rule-based penetration identification**
  - Typically, the rules used in these systems are specific to the machine and operating system
  - The most fruitful approach to developing such rules is to analyze attack tools and scripts collected on the Internet
  - These rules can be supplemented with rules generated by knowledgeable security personnel
- USTAT
  - A model independent of specific audit records
  - Deals in general actions rather than the detailed specific actions recorded by the UNIX auditing mechanism
  - Implemented on a SunOS system that provides audit records on 239 events

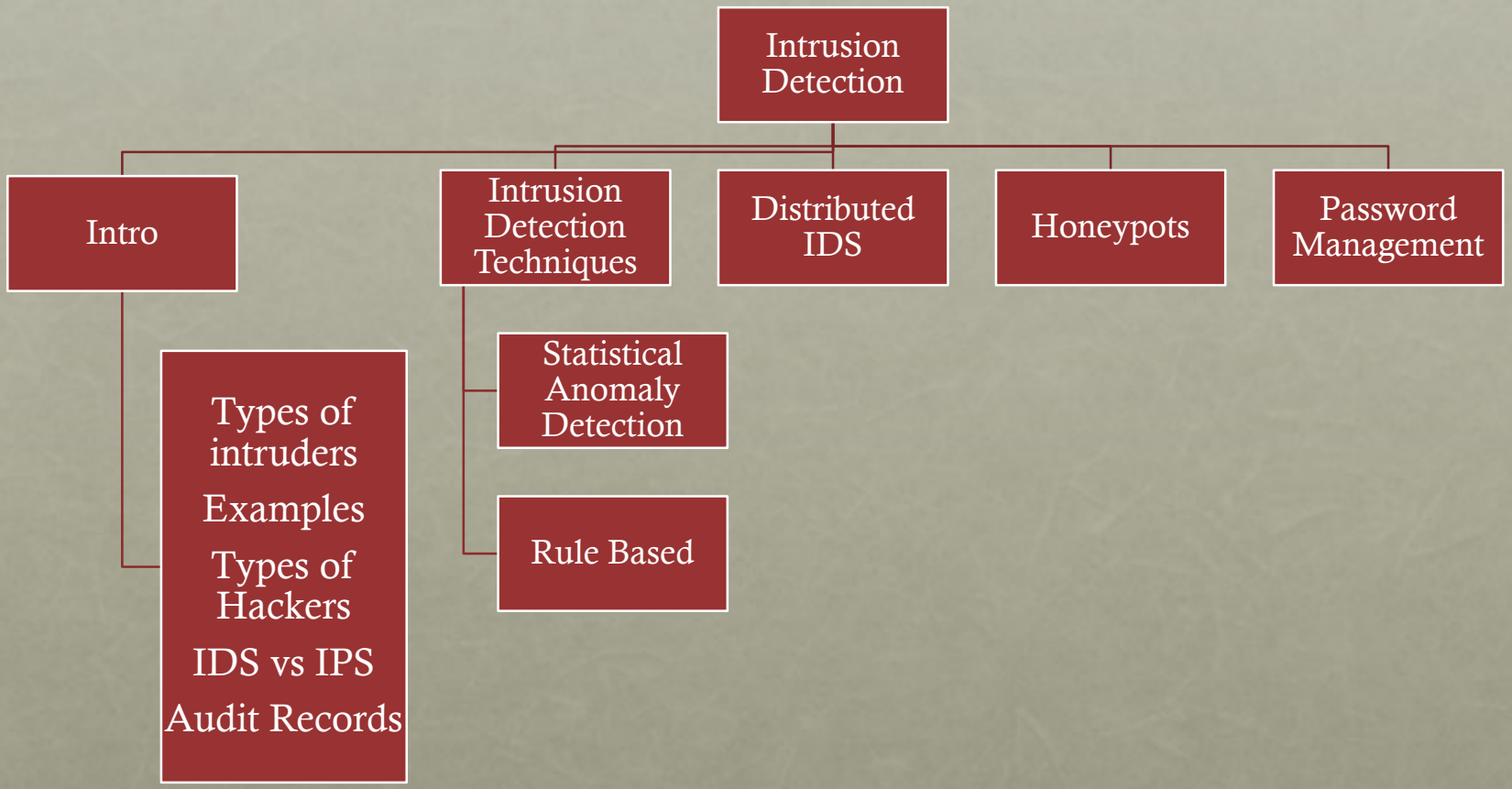
Statistical Anomaly Detection



Rule-Based Anomaly Detection

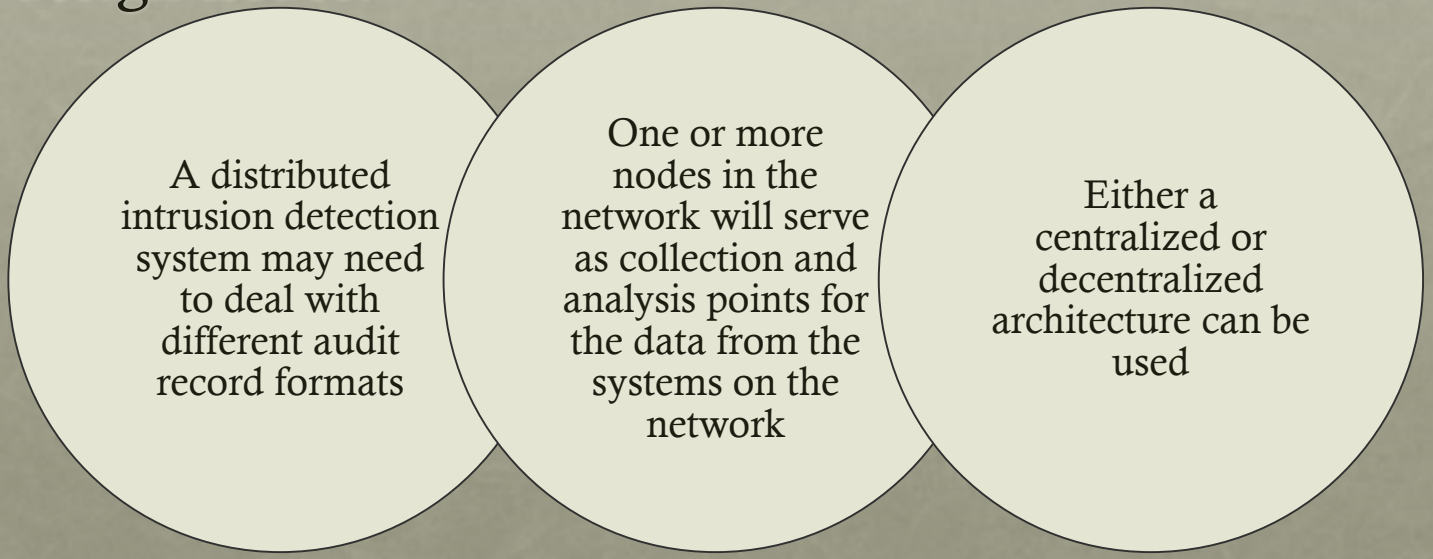


OUTLINE


Intrusion Detection
Intro
Types of intruders
Examples
Types of Hackers
IDS vs IPS
Audit Records
Intrusion Detection Techniques
Statistical Anomaly Detection
Rule Based
Distributed IDS
Honeypots
Password Management

DISTRIBUTED INTRUSION DETECTION

- Traditional systems focused on single-system stand-alone facilities
  - The typical organization, however, needs to defend a distributed collection of hosts supported by a LAN or internetwork
  - A more effective defense can be achieved by coordination and cooperation among intrusion detection systems across the network
- Major design issues:



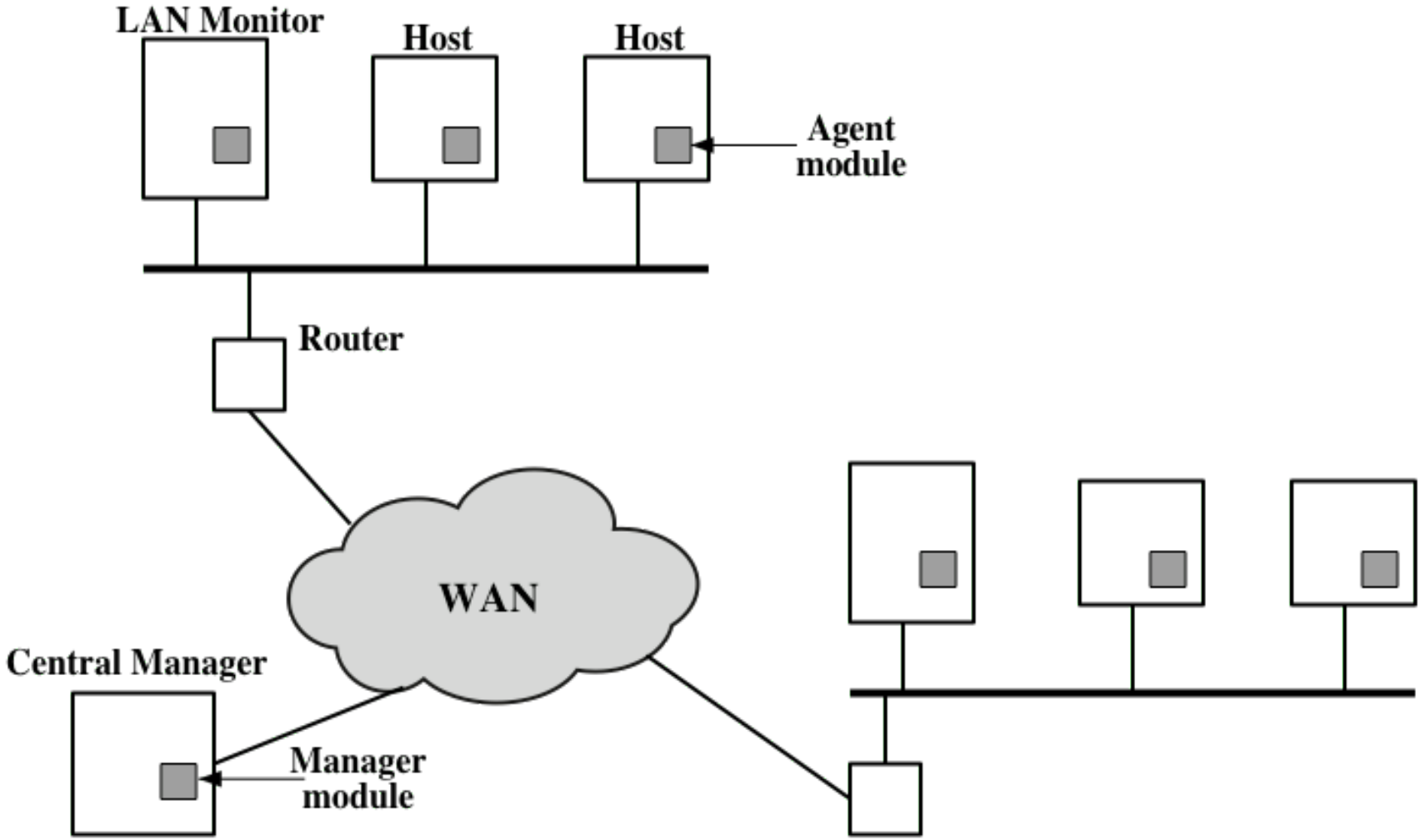


Figure 11.2 Architecture for Distributed Intrusion Detection

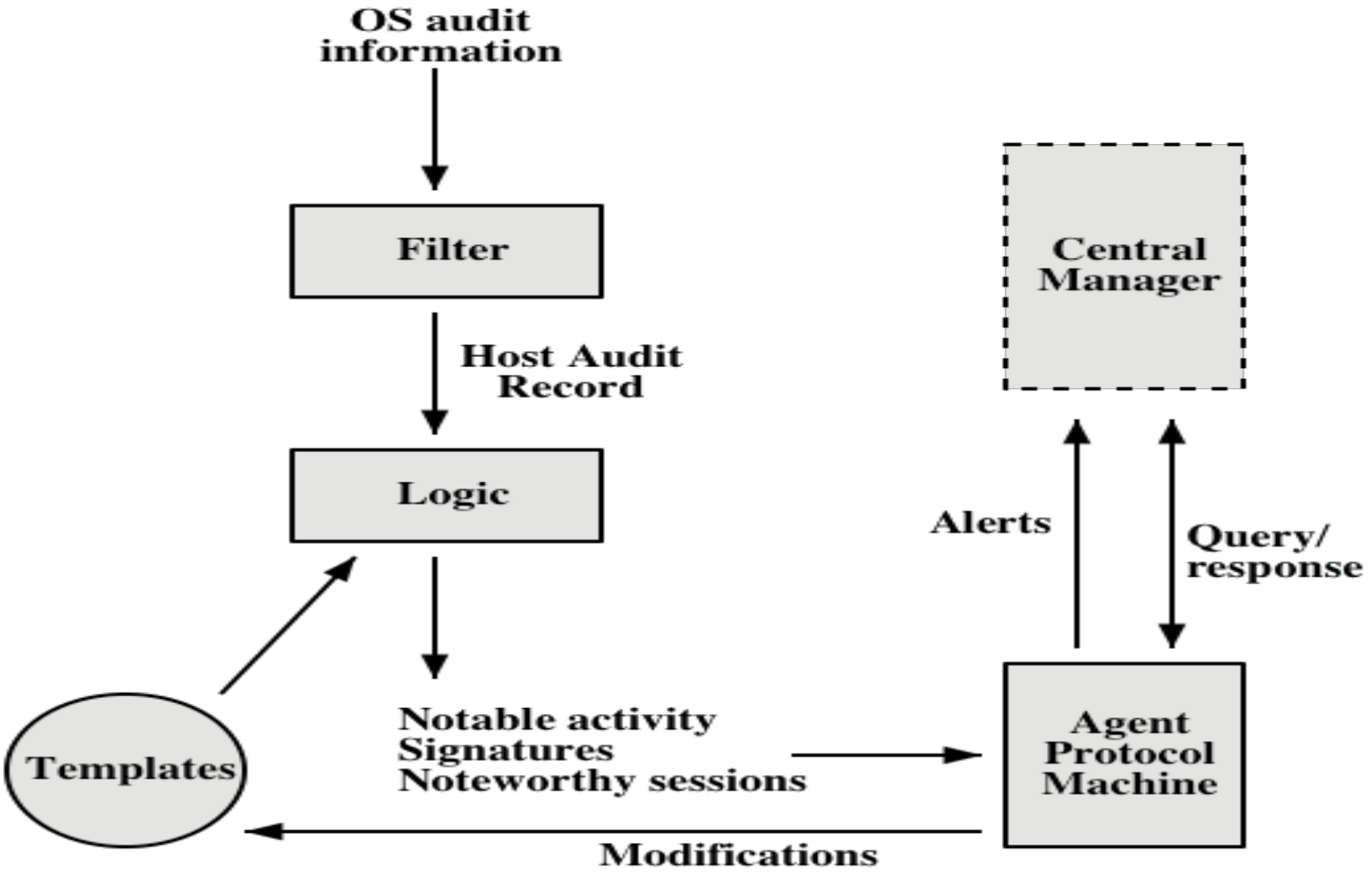


Figure 11.3 Agent Architecture

HONEYPOTS

- Decoy systems that are designed to lure a potential attacker away from critical systems

Has no production value

- These systems are filled with fabricated information designed to appear valuable but that a legitimate user of the system wouldn't access
- Thus, any attempt to communicate with the system is most likely a probe, scan, or attack

Designed to:

- Divert an attacker from accessing critical systems
- Collect information about the attacker's activity
- Encourage the attacker to stay on the system long enough for administrators to respond

- Because any attack against the honeypot is made to seem successful, administrators have time to mobilize and log and track the attacker without ever exposing productive systems

- Recent research has focused on building entire honeypot networks that emulate an enterprise, possible with actual or simulated traffic and data

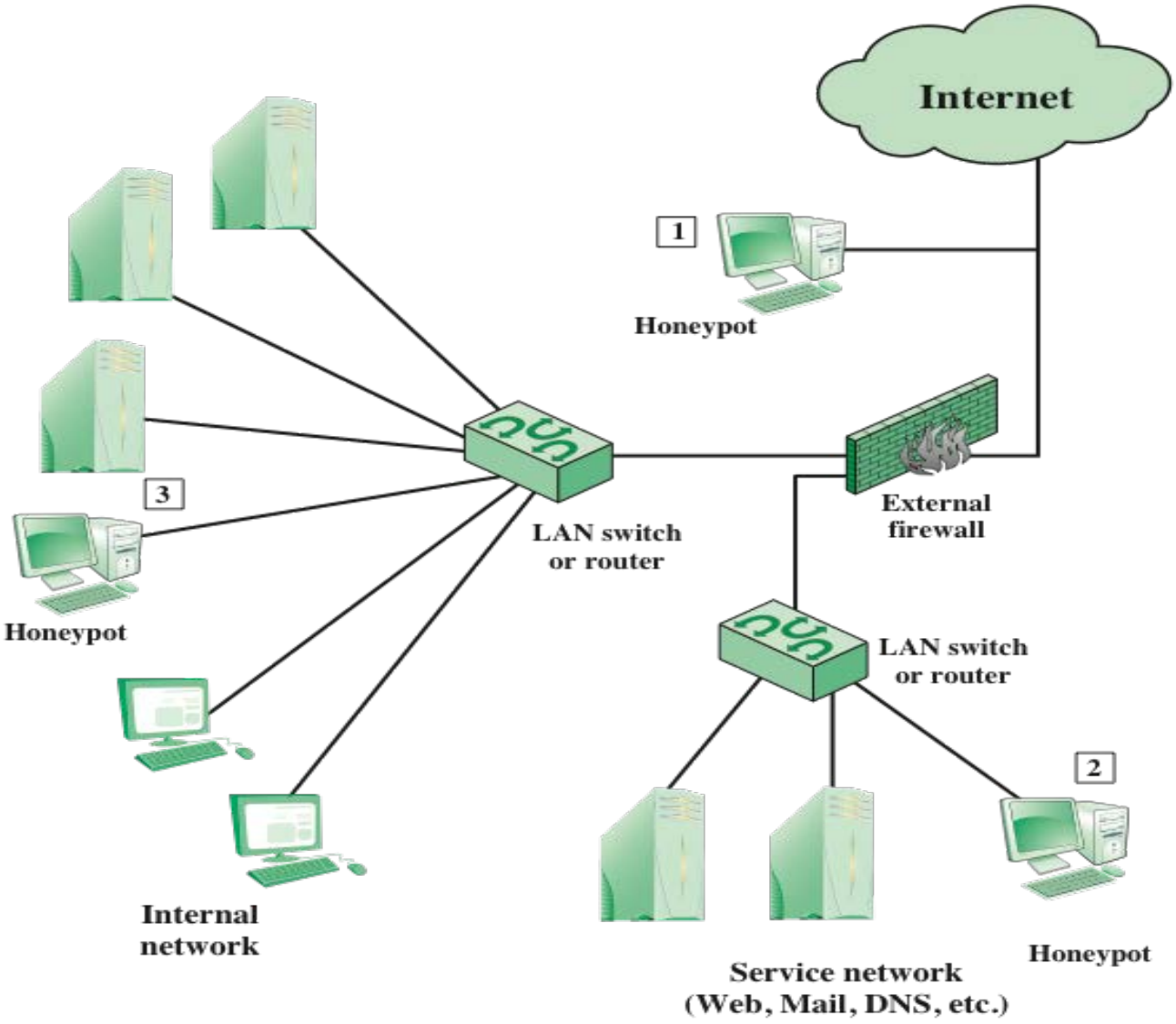


Figure 11.4 Example of Honeypot Deployment

PASSWORD MANAGEMENT

- **Front line of defense against intruders**
- **Virtually all multiuser systems require that a user provide not only a name or identifier (ID) but also a password**
  - **Password serves to authenticate the ID of the individual logging on to the system**
  - **The ID provides security by:**
    - **Determining whether the user is authorized to gain access to a system**
    - **Determining the privileges accorded to the user**
    - **Used in discretionary access control**

ATTACK STRATEGIES AND COUNTERMEASURES

Workstation hijacking

- The attacker waits until a logged-in workstation is unattended
- The standard countermeasure is automatically logging the workstation out after a period of inactivity

Exploiting user mistakes

- Attackers are frequently successful in obtaining passwords by using social engineering tactics that trick the user or an account manager into revealing a password; a user may intentionally share a password to enable a colleague to share files; users tend to write passwords down because it is difficult to remember them
- Countermeasures include user training, intrusion detection, and simpler passwords combined with another authentication mechanism

Offline dictionary attack

- Determined hackers can frequently bypass access controls and gain access to the system's password file
- Countermeasures include controls to prevent unauthorized access to the password file, intrusion detection measures to identify a compromise, and rapid reissuance of passwords should the password file be compromised

Specific account attack

- The attacker targets a specific account and submits password guesses until the correct password is discovered
- The standard countermeasure is an account lockout mechanism, which locks out access to the account after a number of failed login attempts

ATTACK STRATEGIES AND COUNTERMEASURES

Electronic monitoring

- If a password is communicated across a network to log on to a remote system, it is vulnerable to eavesdropping
- Simple encryption will not fix this problem, because the encrypted password is, in effect, the password and can be observed and reused by an adversary

Password guessing against single user

- The attacker attempts to gain knowledge about the account holder and system password policies and uses that knowledge to guess the password
- Countermeasures include training and enforcement of password policies that make passwords difficult to guess

Exploiting multiple password use

- Attacks can become much more effective or damaging if different network devices share the same or a similar password for a given user
- Countermeasures include a policy that forbids the same or similar password on particular network devices

Popular password attack

- Attack is to use a popular password and try it against a wide range of user IDs
- Countermeasures include policies to inhibit the selection by users of common passwords and scanning the IP addresses of authentication requests and client cookies for submission patterns

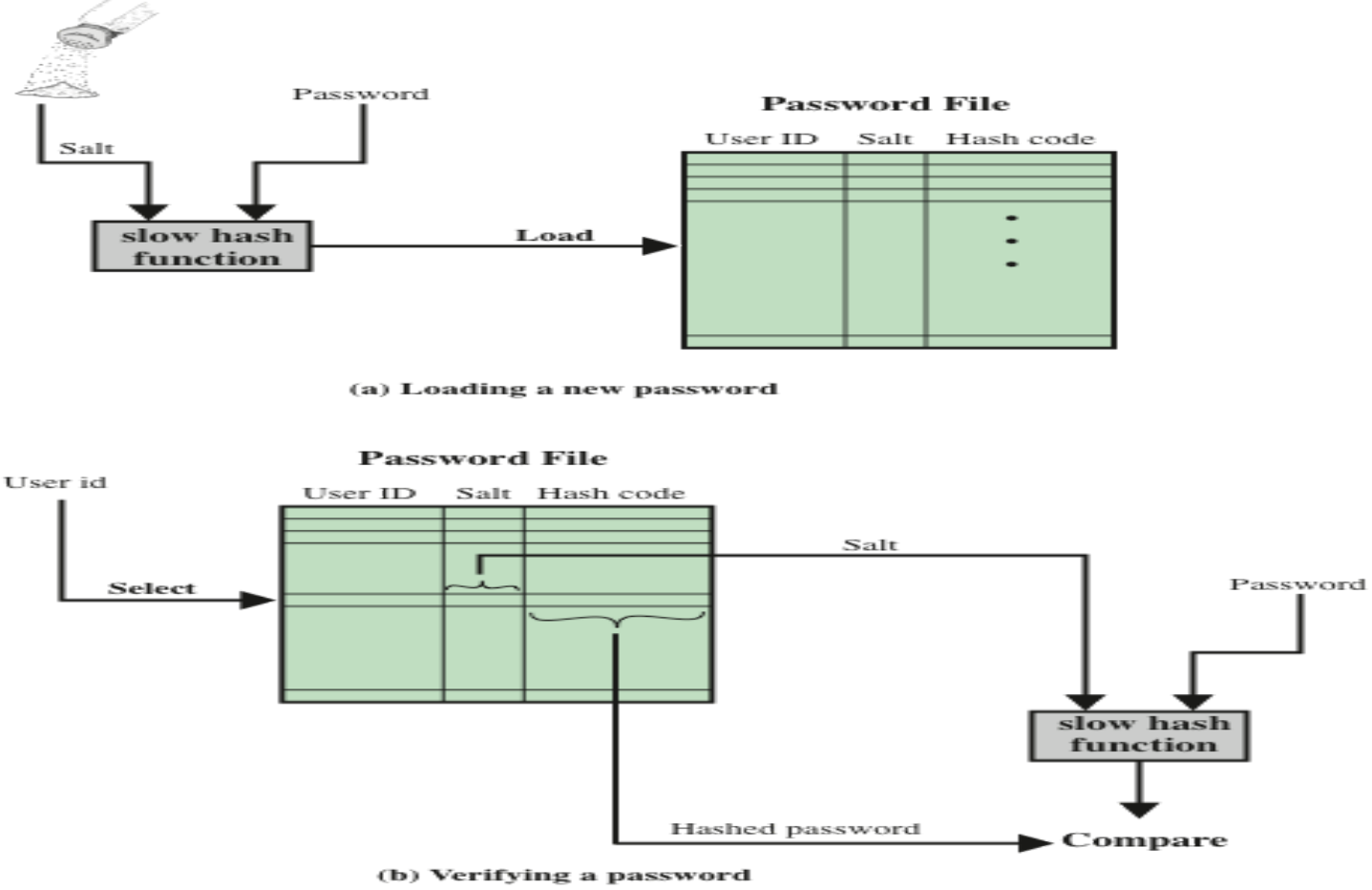


Figure 11.6 UNIX Password Scheme

Benefits of salt value:

1- It greatly increases the difficulty of offline dictionary attacks.

2- It prevents duplicate passwords from being visible in the password file.

3- It becomes nearly impossible to find out whether a person with passwords on two or more systems has used the same password on all of them.

UNIX IMPLEMENTATIONS

- Crypt(3)
  - Was designed to discourage guessing attacks
  - This particular implementation is now considered inadequate
  - Despite its known weaknesses, this UNIX scheme is still often required for compatibility with existing account management software or in multivendor environments
- MD5 secure hash algorithm
  - The recommended hash function for many UNIX systems, including Linux, Solaris, and FreeBSD
  - Far slower than crypt(3)
- Bcrypt
  - Developed for OpenBSD
  - Probably the most secure version of the UNIX hash/salt scheme
  - Uses a hash function based on the Blowfish symmetric block cipher
  - Slow to execute
  - Includes a cost variable

Security Risk Evaluation for Cloud Systems

Outline

- What is risk?
- What is security risk?
- How to compute risk in computer networks?
- Application example

What is risk?

- **Risk**: Probability of losing/damaging something of value
- **Security Risk**: Probability of loss of or damage to an asset resulting from a security threat.
- **Computation/Evaluation**: In general risk is given by

- Risk = Impact (α) * Probability(β)

- **WHERE**
  - α = ***consequence(s)*** *of the bad event e.g. if an attack correctly guesses the PIN code to your bank account, they will steal your money.*
  - β = *the* ***probability*** *of successful exploitation of a vulnerability - leading to an attack e.g. there is a 90% chance that an attacker can correctly guess a weak PIN code.*

- Risk of losing money in a bank account because of a weak PIN code:

$$Risk = QAR100{,}000 * 0.9 = QAR90{,}000$$

Security Risk

In the context of security, a security risk is given by a product of Impact and Exploitability of a vulnerability:

- **Impact:**
  - the consequence of a security attack where a vulnerability has been successfully exploited.
  - Impact is derived from the value of the assets being protected.
- **Exploitability**
  - a measure of how easy it is for an attacker to successfully exploit a vulnerability.
  - For example, it is easier to correctly guess a weak password than it is to guess a strong password.
  - Exploitability is derived from the vulnerabilities in the system.

Security Risk of a Single Host/Node in a Network

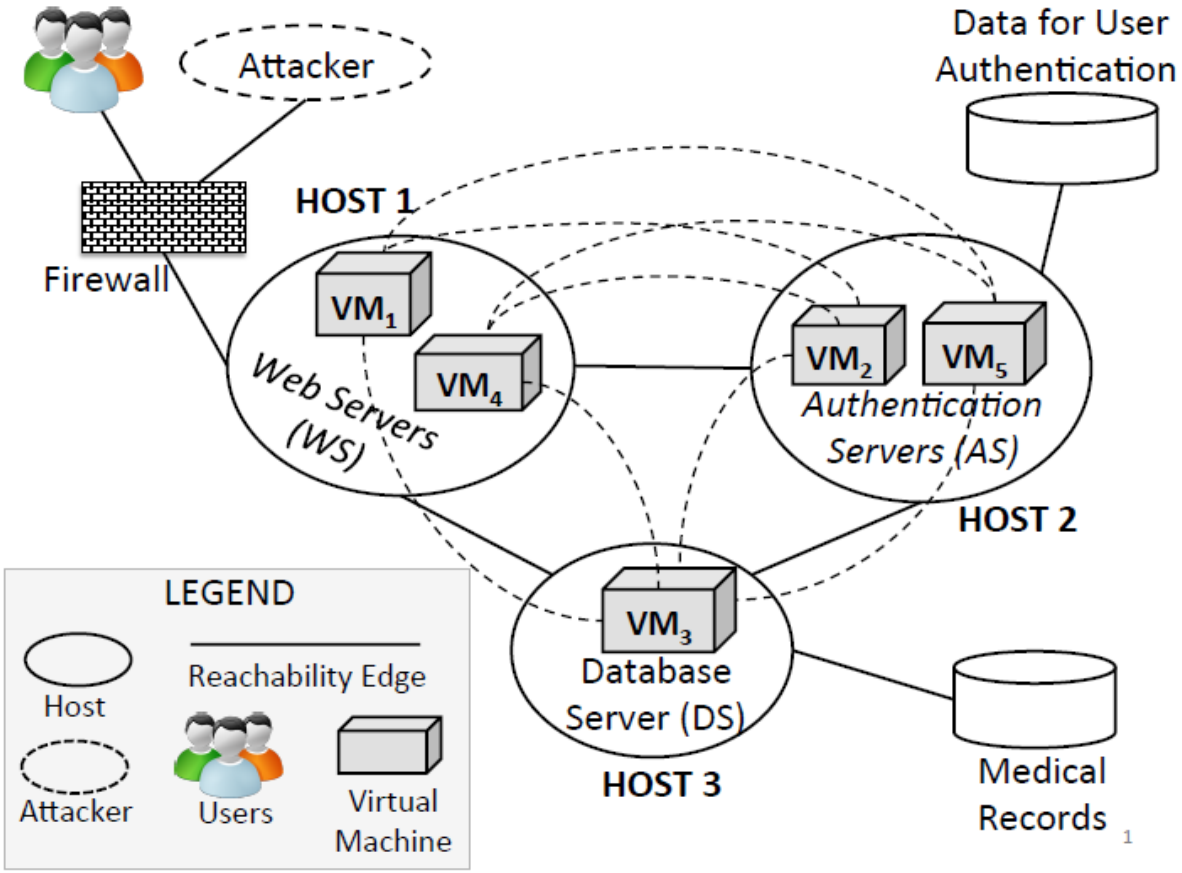
- Given by the sum of the risks with respect to each vulnerability in a host.
- Vulnerability information is obtained through documentation sources such as the following databases
  - CVE Details: https://www.cvedetails.com/
  - NVD: https://nvd.nist.gov/
- Every new vulnerability(v) is recorded in these databases.
- Key Characteristics of a vulnerability:
  - **Product(s)**: This is the set of products/software system where the vulnerability is applicable or that are affected by the this vulnerability e.g. Window 10 Home Edition, Acrobat Reader Version 5.6.2, etc
  - **Exploitability(β) Score** – same as defined in the previous slide
  - **Impact Score(α)** – see previous slide
  - **Threats**: This is the set of threats posed by the vulnerability e.g. a weak password for your Blackboard account poses the threat of Information Disclosure(I).

- The security risk of host A is: $Risk_A = \sum_{j=1}^{N} (\alpha_j * \beta_j)$

STRIDE threat modelling framework

	Threat	Property Violated	Threat Definition
S	Spoofing identify	Authentication	Pretending to be something or someone other than yourself
T	Tampering with data	Integrity	Modifying something on disk, network, memory, or elsewhere
R	Repudiation	Non-repudiation	Claiming that you didn't do something or were not responsible; can be honest or false
I	Information disclosure	Confidentiality	Providing information to someone not authorized to access it
D	Denial of service	Availability	Exhausting resources needed to provide service
E	Elevation of privilege	Authorization	Allowing someone to do something they are not authorized to do

Medical Records System Model: An example



VM	OS	VULNERABILITIES			
		CVE-ID	α (Imp.)	β (Prob.)	γ (*Threats*)
VM_1	WIN10	CVE-2017-0280	3.6	0.22	D
VM_2	WIN10	CVE-2017-0280	3.6	0.22	D
		CVE-2017-0246	5.9	1.00	T,I,D
VM_3	RHEL	CVE-2017-1000376	5.9	1.00	T,I,D
VM_4	Linux	CVE-2017-10810	3.6	0.39	D
VM_5	Linux	CVE-2017-1297	5.9	0.13	T,I,D

Real-time vulnerability data from NVD database

$$\lambda(n_i) = \sum_{j=1}^{|V(n_i)|} \alpha_{v_j}, v_j \in V(n_i), V(n_i) \subseteq \mathbb{V}$$

$$\mu(n_i) = 1 - \prod_{j=1}^{|V(n_i)|} \{1 - \beta_{v_j}\}, v_j \in V(n_i), V(n_i) \subseteq \mathbb{V}.$$

The impact on a component is the sum of impact of all its vulnerabilities

The exploitability of a component: the combined probability of failing to exploit all the vulnerabilities.

$$P_{exploitability} = 1 - P_{No-exploitability}$$

Each component has several vulnerabilities
Each vulnerability has a probability of $\beta_1, \beta_2, \ldots, \beta_k$
The probability of no-exploitability of vulnerability 1 is $(1 - \beta_1)$
The probability of no-exploitability of vulnerability 2 is $(1 - \beta_2)$
The probability of no-exploitability of vulnerability k is $(1 - \beta_k)$

$$P_{No-expl\ of\ Vun\ 1} = 1 - \beta_1$$

$$P_{No-expl\ of\ Vun\ 2} = 1 - \beta_2$$

$$P_{No-expl\ of\ Vun\ k} = 1 - \beta_k$$

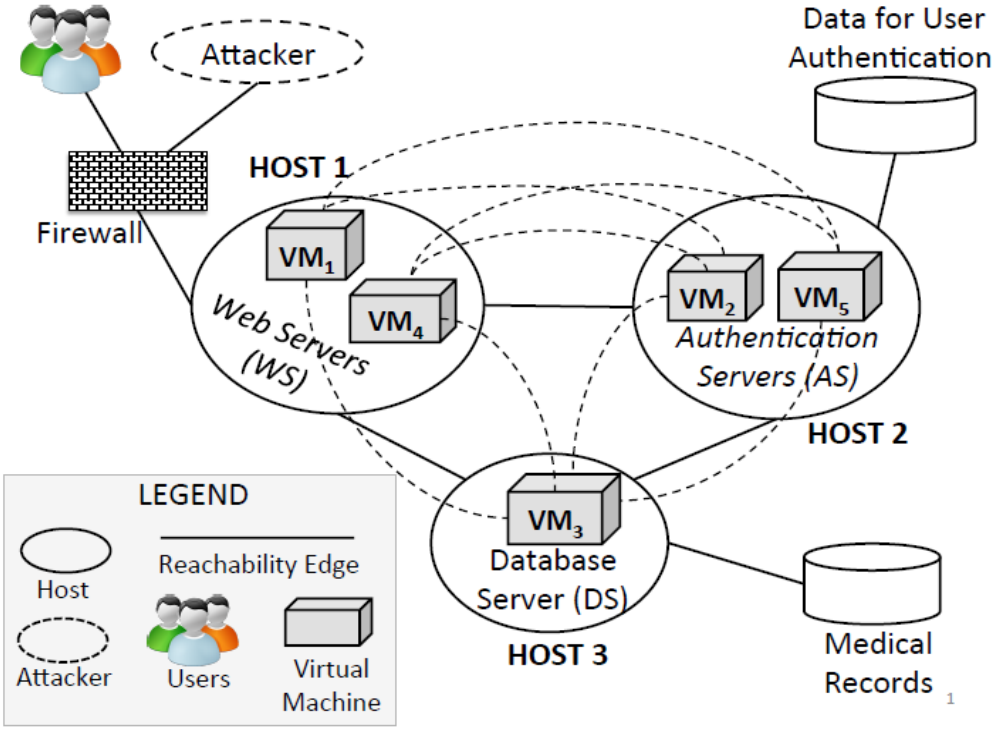
$$P_{No-exploitability} = (1 - \beta_1) * (1 - \beta_2) * \cdots * (1 - \beta_k)$$

$$P_{No-exploitability} = \prod_{i=1}^{k} 1 - \beta_i$$

$$\mu_{exploitability} = 1 - \prod_{i=1}^{k} 1 - \beta_i$$

Threat Weighting

Weighting Assignment depends on **Security Requirement** and **Function** of VM.



THREAT IMPACT WEIGHTING ASSIGNMENT TABLE

VM	$\omega(n_i, t)$					
	S	T	R	I	D	E
VM_1	0.40	0.00	0.00	0.00	0.60	0.00
VM_2	0.05	0.20	0.00	0.25	0.00	0.50
VM_3	0.00	0.20	0.00	0.80	0.00	0.00
VM_4	0.40	0.00	0.00	0.00	0.60	0.00
VM_5	0.05	0.20	0.00	0.25	0.00	0.50

Risk Evaluation

VM	OS	VULNERABILITIES			
		CVE-ID	α (Imp.)	β (Prob.)	γ (*Threats*)
VM_1	WIN10	CVE-2017-0280	3.6	0.22	D
VM_2	WIN10	CVE-2017-0280	3.6	0.22	D
		CVE-2017-0246	5.9	1.00	T,I,D
VM_3	RHEL	CVE-2017-1000376	5.9	1.00	T,I,D
VM_4	Linux	CVE-2017-10810	3.6	0.39	D
VM_5	Linux	CVE-2017-1297	5.9	0.13	T,I,D

THREAT IMPACT WEIGHTING ASSIGNMENT TABLE

VM	$\omega(n_i, t)$					
	S	*T*	*R*	*I*	*D*	*E*
VM_1	0.40	0.00	0.00	0.00	0.60	0.00
VM_2	0.05	0.20	0.00	0.25	0.00	0.50
VM_3	0.00	0.20	0.00	0.80	0.00	0.00
VM_4	0.40	0.00	0.00	0.00	0.60	0.00
VM_5	0.05	0.20	0.00	0.25	0.00	0.50

RISKS BEFORE ELIMINATION OF IRRELEVANT THREATS

VM	$\lambda(n_i)$	$\mu(n_i)$	$Risk(n_i, \kappa)$						$Risk(n_i)$
			S	*T*	*R*	*I*	*D*	*E*	
VM_1	3.60	0.22	0.32	0.00	0.00	0.00	0.48	0.00	**0.80**
VM_2	9.50	1.00	0.48	1.90	0.00	2.38	0.00	4.75	**9.51**
VM_3	5.90	1.00	0.00	1.18	0.00	4.72	0.00	0.00	**5.90**
VM_4	3.60	0.39	0.56	0.00	0.00	0.00	0.84	0.00	**1.40**
VM_5	5.90	0.13	0.04	0.15	0.00	0.19	0.00	0.38	**0.76**
			1.39	**3.23**	**0.00**	**7.29**	**1.32**	**5.13**	**18.37**

Threat Relevance Determination

VM	OS	VULNERABILITIES				
		CVE-ID		α (Imp.)	β (Prob.)	γ (*Threats*)
VM_1	WIN10	CVE-2017-0280		3.6	0.22	D
VM_2	WIN10	CVE-2017-0280		3.6	0.22	D
		CVE-2017-0246		5.9	1.00	T,I,D
VM_3	RHEL	CVE-2017-1000376		5.9	1.00	T,I,D
VM_4	Linux	CVE-2017-10810		3.6	0.39	D
VM_5	Linux	CVE-2017-1297		5.9	0.13	T,I,D

RISKS AFTER ELIMINATION OF IRRELEVANT THREATS

VM	$\lambda(n_i)$	$\mu(n_i)$	$Risk(n_i, \kappa)$						$Risk(n_i)$
			S	*T*	*R*	*I*	*D*	*E*	
VM_1	3.60	0.22	**0.00**	0.00	0.00	0.00	0.48	0.00	**0.48**
VM_2	9.50	1.00	**0.00**	1.90	0.00	2.38	0.00	**0.00**	**4.28**
VM_3	5.90	1.00	0.00	1.18	0.00	4.72	0.00	0.00	**5.90**
VM_4	3.60	0.39	**0.00**	0.00	0.00	0.00	0.84	0.00	**0.84**
VM_5	5.90	0.13	**0.00**	0.15	0.00	0.19	0.00	0.00	**0.34**
			0.00	**3.23**	**0.00**	**7.29**	**1.32**	**0.00**	**11.84**

Steps

- Calculate risk per VM
  - If one vulnerability per VM → risk = α * β
  - More than one vulnerability per VM,
    - sum α,
    - then use the product formula to calculate β
    - Then multiply
- STRIDE Weights
  - Multiply threat weight (percentage) by calculated risk per VM
  - Sum all rows to calculate the risk associated with each threat
- If irrelevant threat are eliminated, set corresponding weight to 0 and recalculate

SUMMARY

- Intruders
  - Behavior patterns
  - Intrusion techniques
- Intrusion detection
  - Audit records
  - Statistical anomaly detection
  - Rule-based intrusion detection
  - The base-rate fallacy
  - Distributed intrusion detection
  - Honeypots
  - Intrusion detection exchange format
- Password management
  - The vulnerability of passwords
  - The use of hashed passwords
  - User password choices
- Security Risk Evaluation
  - What is security risk
  - How to compute risk
  - Medical Records System example