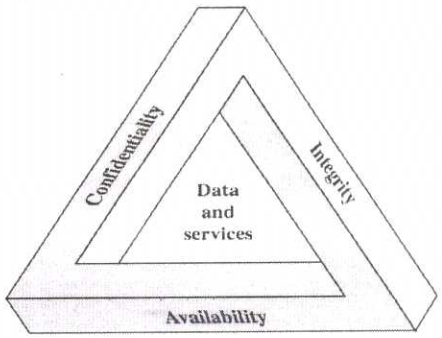
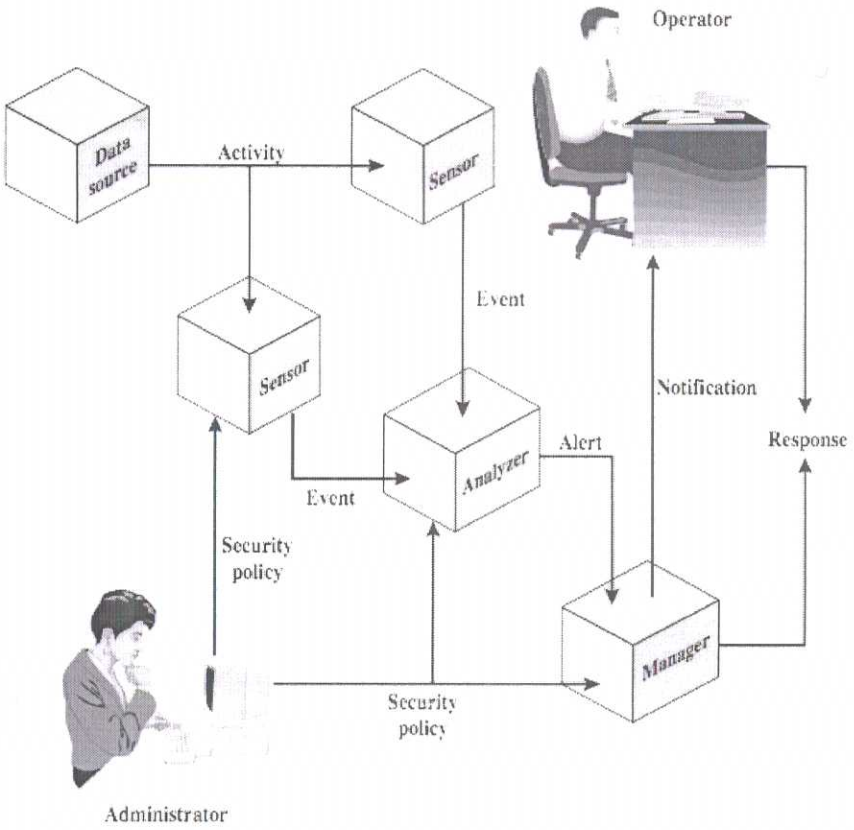
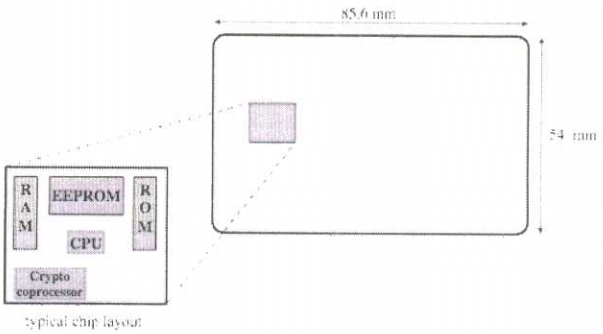
	Information Security - Rev 2015 TED 5136 ANSWER KEY		
Q no	**Scoring Indicator**	**split score**	**Total score**
I	**PART A**		
1	**Authenticity:** The property of being genuine and being able to be verified and trusted; confidence in the validity of a transmission, a message, or message originator.	2	2
2	Specification,implementation,correctness	Any 2	2
3	**Masquerader:** An individual, likely an outsider, not authorized to use the computer and who penetrates a system's access controls to exploit a legitimate user's account	2	2
4	**Firewalls** can be an effective means of protecting a local system or network of systems from network-based security threats while at the same time affording access to the outside world via wide area networks and the Internet. Internet connectivity is no longer optional for organizations. The information and services available are essential to the organization	2	2
5	**Interception:** An unauthorized entity directly accesses sensitive data in transit.	2	2
II	**PART B**		
1	computer security triad  • **Confidentiality:** Preserving authorized restrictions on information access and disclosure, including means for protecting personal privacy and proprietary information. A loss of confidentiality is the unauthorized disclosure of information. • **Integrity:** Guarding against improper information modification or destruction, and includes ensuring information non-repudiation and authenticity. A loss of integrity is the unauthorized modification or	3 points —2 marks each	6

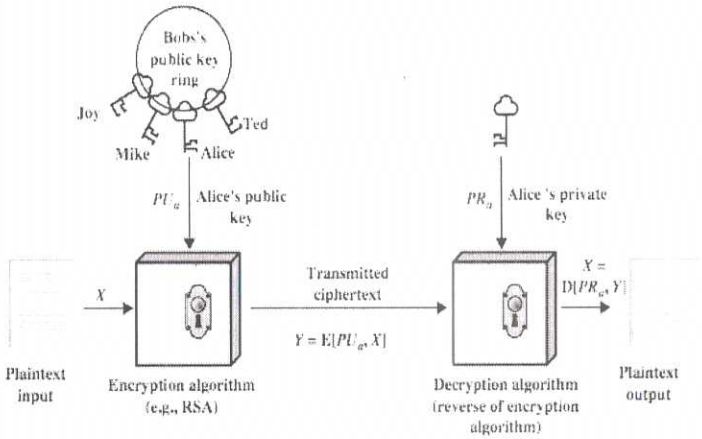
	destruction of information. • **Availability:** Ensuring timely and reliable access to and use of information. A loss of availability is the disruption of access to or use of information or an information system		
2	**Access Control Elements** The basic elements of access control are: subject, object, and access right. A **subject** is an entity capable of accessing objects, usually a process. Any user or application actually gains access to an object by means of a process that represents it. A subject is typically held accountable for the actions they have initiated, and an audit trail may be used to associate with a subject and security-relevant actions performed on an object. Basis access control systems typically define three classes of subject: 1•**Owner:** This may be the creator of a resource, such as a file. For system resources, ownership may belong to a system administrator. For project resources, a project administrator or leader my be assigned ownership. 2• **Group:** In addition to the privileges assigned to an owner, a named group of users may also be granted access rights, such that membership in the group is sufficient to exercise these access rights. 3• **World:** The least amount of access is granted to users who are able to access the system but are not included in the categories owner and group for this resource. An **object** is any resource to which access is controlled. In general, and object is an entity used to contain and/or receive information. Examples include records, blocks, pages, segments, files, portions of files, directories, directory trees, mailboxes, messages, and programs. The number and types of objects to be protected by an access control system depends on the environment in which access control. An **access right** describes the way in which a subject may access an object. Access rights could include the following: read, write, execute, delete, create, search.	3 points —2 marks each	6

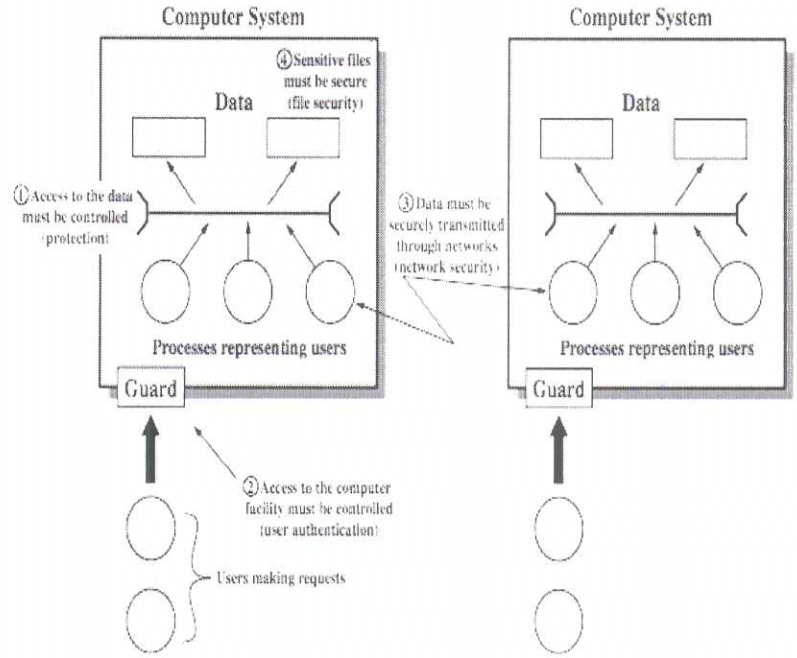
3	Intrusion Detection Exchange Format  The specification defines formats for event and alter messages, message types, and exchange protocols for communication of intrusion detection information. The functional components are as follows: • Data source: raw data an IDS uses to detect unauthorized or undesired activity • Sensor: collects data from the data source & forwards events to the analyzer • Analyzer: process analyzing data collected for unauthorized/undesired activity • Administrator: human with overall responsibility for setting security policy of org • Manager: process from which operator manages components of ID system • Operator: human that is the primary user of the IDS manager	Fig—3 Expln --3	6

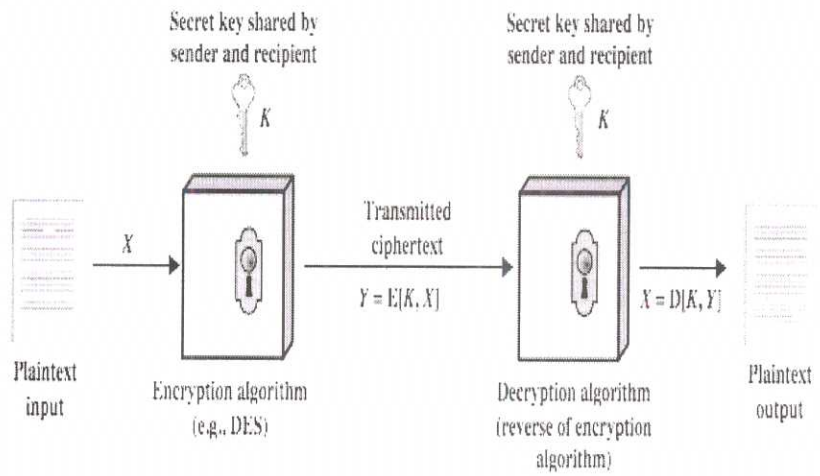
	The sensor monitors data sources looking for suspicious activity. The sensor communicates suspicious activity to the analyzer as an event. If the analyzer determines that the event is of interest, it sends an alert to the manager component. The manager component issues a notification to the human operator. A response can be initiated automatically by the manager component or by the human operator. The security policy is the predefined, formally documented statement that defines what activities are allowed to take place.		
4	**Bastion Hosts** A bastion host is a system identified by the firewall administrator as a critical strong point in the network's security. Typically, the bastion host serves as a platform for an application-level or circuit-level gateway. Common characteristics of a bastion host: • executes a secure version of its operating system, making it a trusted system. • only essential services are installed on the bastion host. These include proxy applications such as Telnet, DNS, FTP, SMTP, and user authentication. • may require additional authentication before a user is allowed access to the proxy services, and may require its own authentication before granting user access. • each proxy is configured to support only a subset of the application's command set. • each proxy is configured to allow access only to specific host systems. • each proxy maintains detailed audit information by logging all traffic, each connection, and the duration of each connection. • each proxy module is a very small software package specifically designed for network security, hence is easier to check such modules for security flaws. • each proxy is independent of other proxies on the bastion host, and can be uninstalled without affecting the operation of the other proxy applications. • generally performs no disk access other than to read its initial configuration file. • each proxy runs as a nonprivileged user in a private and secured directory on host.	Any 6 points	6

5	**Token Authentication** ➢ object user possesses to authenticate, e.g. ● embossed card- embosed character.eg old credit card ● magnetic stripe card-ATM ● memory card-prepaid phone ● Smartcard-biometric id **Memory Card** ➢ store but do not process data ➢ magnetic stripe card, e.g. bank card ➢ electronic memory card ➢ used alone for physical access ➢ with password/PIN for computer use ➢ drawbacks of memory cards include: ● need special reader ● loss of token issues ● user dissatisfaction **Smartcard**  typical chip layout ➢ credit-card like ➢ has own processor, memory, I/O ports ● wired or wireless access by reader ● may have crypto co-processor ● ROM, EEPROM, RAM memory ➢ executes protocol to authenticate with reader/computer	Any 2—3 marks each	6
6	IDS Requirements • run continually with minimal human supervision. • be fault tolerant in the sense that it must be able to recover from system crashes and reinitializations. • resist subversion. The IDS must be able to monitor itself and detect if it has been modified by an attacker. • impose a minimal overhead on the system where it is running. • be able to be configured according to the security policies of the system that is being monitored. • be able to adapt to changes in system and user behavior over time.	Any 6 points	

	• be able to scale to monitor a large number of hosts. • provide graceful degradation of service in the sense that if some components of the IDS stop working for any reason, the rest of them should be affected as little as possible. • allow dynamic reconfiguration; that is, the ability to reconfigure the IDS without having to restart it.		6
7	**Types of Flooding Attacks** Flooding attacks take a variety of forms, based on which network protocol is being used to implement the attack. Common flooding attacks use any of the ICMP, UDP or TCP SYN packet types. An ICMP flooding attack uses an **ICMP packet**, such as ICMP echo request packets in a ping flood. This type of ICMP packet was chosen since traditionally network administrators allowed such packets into their networks. More recently, many organizations have restricted the ability of these packets to pass through their firewalls. In response, attackers have started using other ICMP packet types. Since some of these should be handled to allow the correct operation of TCP/IP, they are much more likely to be allowed through an organization's firewall An alternative to using ICMP packets is to **use UDP packets** directed to some port number, and hence potential service, on the target system. Spoofed source addresses are normally used if the attack is generated using a single source system, for the same reasons as with ICMP attacks. Another alternative is to send **TCP packets** to the target system. Most likely these would be normal TCP connection requests, with either real or spoofed source addresses. In this case, it is the total volume of packets that is the aim of the attack, rather than specifically targeting the system code. This is the difference between a SYN spoofing attack and a SYN flooding attack.	3 attack —2 marks each	6
	PART-C		

III (a)	**Public Key Encryption**  (a) Confidentiality . A public-key encryption scheme has six ingredients, as shown here in Figure 2.6a: • **Plaintext:** the readable message or data that is fed into the algorithm as input. • **Encryption algorithm:** performs various transformations on the plaintext. • **Public and private key:** a pair of keys selected so that if one is used for encryption, the other is used for decryption. The exact transformations performed by the encryption algorithm depend on the public or private key that is provided as input. • **Ciphertext:** the scrambled message produced as output that depends on the plaintext and key. For a given message, two different keys produce two different ciphertexts. • **Decryption algorithm:** takes ciphertext and key to produces the original plaintext. As the names suggest, the public key of the pair is made public for others to use, while the private key is known only to its owner. A public-key cryptographic algorithm relies on one key for encryption and a different but related key for decryption. All participants have access to public keys, and private keys are generated locally by each participant and therefore need never be distributed. As long as a user protects his or her private key, incoming communication is secure.	Fig—4 Expln --4	8

(b)	**Scope of Computer Security**  **Hardware** - A major threat = is the threat to availability. Hardware is the most vulnerable to attack and the least susceptible to automated controls. Threats include accidental and deliberate damage to equipment as well as theft. Theft of CDROMs and DVDs can lead to loss of confidentiality. Physical and administrative security measures are needed to deal with these threats. **Software** - includes the operating system, utilities, and application programs. A key threat is an attack on availability. Software is often easy to delete. Software can also be altered or damaged to render it useless. Careful software configuration management can maintain high availability. A more difficult problem is software modification (e.g. from virus/worm) that results in a program that still functions but that behaves differently than before, which is a threat to integrity/authenticity. **Data** - involves files and other forms of data controlled by individuals, groups, and business organizations. Security concerns with respect to data are broad, encompassing availability, secrecy, and integrity. In the case of availability, the concern is with the destruction of data files, which can occur either accidentally or maliciously. The obvious concern with secrecy is the unauthorized reading of data files or databases. A less obvious secrecy threat involves the analysis of data and manifests itself in the use of so-called statistical databases, which provide summary or aggregate information. Finally, data integrity is a major concern in most installations. Modifications to data files can have consequences ranging from minor to disastrous.	Fig--3.5 Expln —3.5	7

IV (a)	**Symmetric Encryption**  A symmetric encryption scheme has five ingredients, as shown here in Figure 2.1 from the text. • **Plaintext:** This is the original message or data that is fed into the algorithm as input. • **Encryption algorithm:** The encryption algorithm performs various substitutions and transformations on the plaintext. • **Secret key:** The secret key is also input to the encryption algorithm. The exact substitutions and transformations performed by the algorithm depend on the key. • **Ciphertext:** This is the scrambled message produced as output. It depends on the plaintext and the secret key. For a given message, two different keys will produce two different ciphertexts. • **Decryption algorithm:** This is essentially the encryption algorithm run in reverse. It takes the ciphertext and the secret key and produces the original plaintext. There are two requirements for secure use of symmetric encryption: 1. We need a strong encryption algorithm. **2.** Sender and receiver must have secure obtained, & keep secure, the secret key.	Fig—4 Expln --4	8

(b)	**Security Functional Requirements** ➢ technical measures: ● access control; identification & authentication; system & communication protection; system & information integrity ➢ management controls and procedures ● awareness & training; audit & accountability; certification, accreditation, & security assessments; contingency planning; maintenance; physical & environmental protection; planning; personnel security; risk assessment; systems & services acquisition ➢ overlapping technical and management: ● configuration management; incident response; media protection	7	7
V (a)	**Biometric Authentication** A biometric authentication system attempts to authenticate an individual based on unique physical characteristics.These include static characteristics, such as fingerprints, hand geometry, facial characteristics, and retinal and iris patterns; and dynamic characteristics, such as voiceprint and signature. Compared to passwords and tokens, biometric authentication is both technically complex and expensive, and have yet to mature as a standard tool for user authentication to computer systems. Figure 3.6 from the text gives a rough indication of the relative cost and accuracy of the most common biometric measures: • Facial characteristics: define characteristics based on relative location and shape of key facial features, such as eyes, eyebrows, nose, lips, and chin shape. • Fingerprints: the pattern of ridges and furrows on the surface of the fingertip, believed to be unique across the entire human population. Automated fingerprint systems extract a number of features to use as a surrogate for the full pattern. • Hand geometry: identify features of hand,: e.g. shape, lengths & widths of fingers. • Retinal pattern: formed by veins beneath the retinal surface is unique and therefore suitable for identification. Uses a digital image of the	Fig—4 Expn--4	8

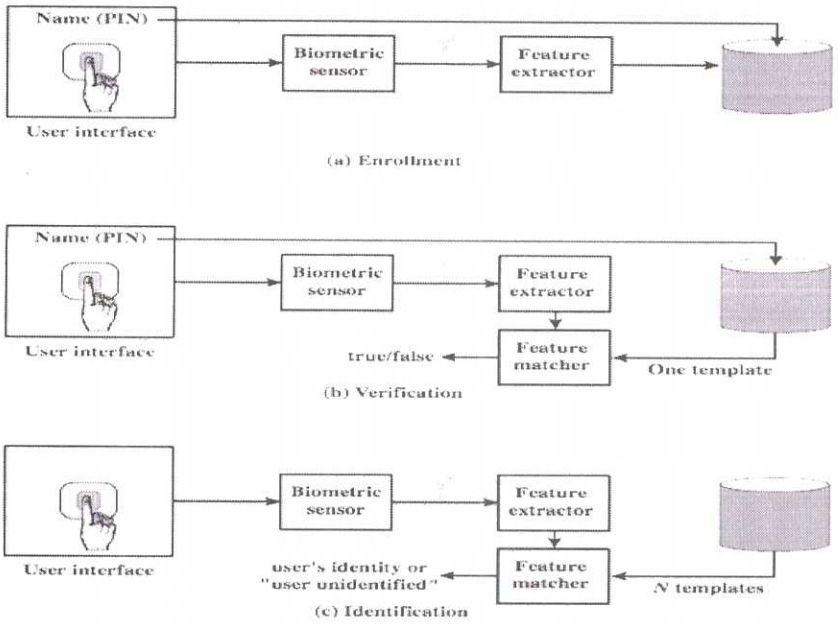
retinal pattern by projecting a low-intensity beam of visual or infrared light into the eye.

• Iris: Another unique physical characteristic is the detailed structure of the iris.

• Signature: each individual has a unique style of handwriting, esp in signature.

• Voice: patterns are more closely tied to physical and anatomical characteristics of the speaker, but still have a variation from sample to sample over time from the same speaker,complicating the biometric recognition task.

Operation of a Biometric System

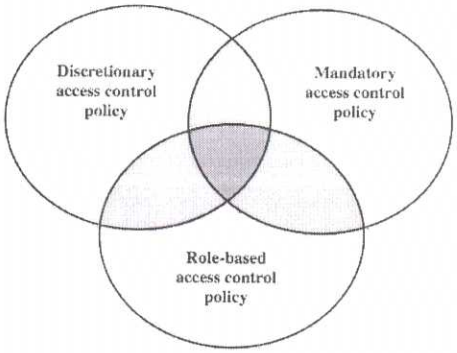


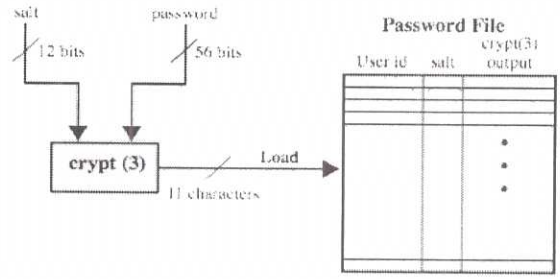
Enrollment

- User enters their name
- Biometric features of users are extracted using biometric sensor
- These features are converted into set of numbers(user template)
- Name, features ,pin are stored in memory

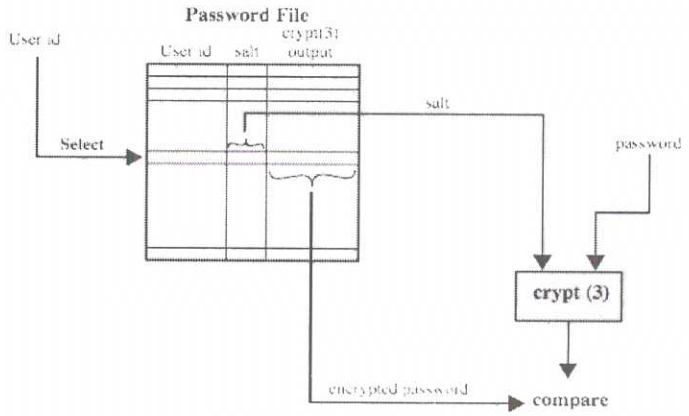
Verification

	➢ First user enters into user interface and their features are extracted ➢ If extracted features matches to stored feature, then user can login **Identification** ➢ User uses only sensor and no additional information is given ➢ Present feature is compared with all the stored feature ➢ If it matches the user is identified.		
(b)	**Password attack strategies** Offline dictionary attack: A determined hacker may bypass access controls and gain access to the system password file. The attacker then compares the password hashes against hashes of commonly used passwords. • Specific account attack: The attacker targets a specific account and submits password guesses until the correct password is discovered. • Popular password attack: The attacker chooses a popular password and try it against a wide range of user IDs. • 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. • Workstation hijacking; The attacker waits until a logged-in workstation is unattended. • Exploiting user mistakes: If the system assigns a password, then the user is more likely to write it down because it is difficult to remember. • Exploiting multiple password use. When different network devices share the same or a similar password for a given user. • Electronic monitoring: If a password is communicated across a network to log on to a remote system, it is vulnerable to eavesdropping.	Any 7	7

VI (a)	**Access Control Policies**  **Discretionary access control (DAC):** based on the identity of the requestor and on access rules (authorizations) stating what requestors are (or are not) allowed to do. This policy is termed *discretionary* because an entity might have access rights that permit the entity, by its own volition, to enable another entity to access some resource. • **Mandatory access control (MAC):** based on comparing security labels (which indicate how sensitive or critical system resources are) with security clearances (which indicate system entities are eligible to access certain resources). This policy is termed *mandatory* because an entity that has clearance to access a resource may not, just by its own volition, enable another entity to access that resource. See chapter 10. • **Role-based access control (RBAC):** based on the roles that users have within the system and on rules stating what accesses are allowed to users in given roles. DAC is the traditional method of implementing access control. MAC is a concept that evolved out of requirements for military information security and is best covered in the context of trusted systems. RBAC has become increasingly popular. These three policies are not mutually exclusive as shown in Figure 4.2. Can employ two or even all three of these policies to cover different classes of system resources.	8	8
(b)	**Use of Hashed Passwords**	Fig—3.5 Expn-3.5	7

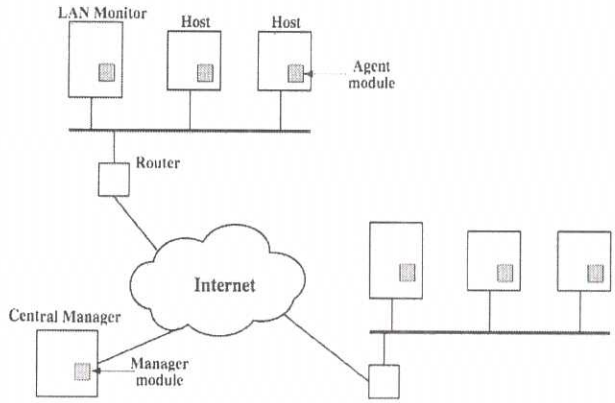


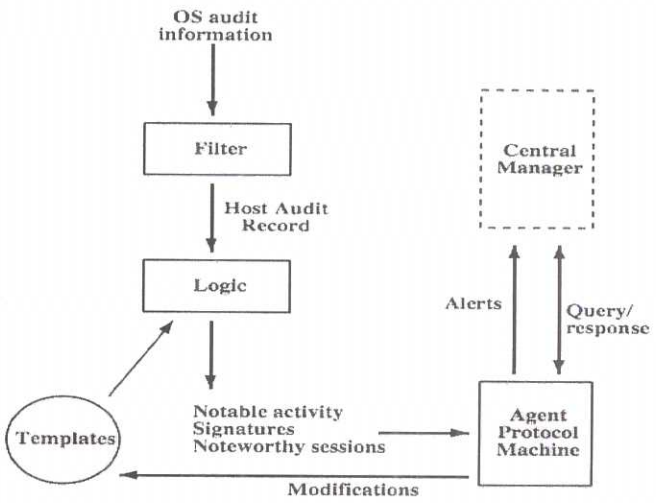
(a) Loading a new password



(b) Verifying a password

A widely used password security technique is the use of hashed passwords and a salt value. This scheme is found on virtually all UNIX variants as well as on a number of other operating systems. The procedure shown here in Figure 3.1a from the text is used. To load a new password into the system, the user selects or is assigned a password.This password is combined with a fixed-length salt value In older implementations, the salt is related to the time the password is assigned to the user. Newer implementations use a pseudorandom or random number. The password and salt serve as inputs to a hashing algorithm to produce a fixed-length hash code. The hash algorithm is designed to be slow to execute to thwart attacks. The hashed password is then stored, together with a plaintext copy of the salt, in the password file for the corresponding user ID. The hashed-password method has been shown to be secure against a variety of cryptanalytic attacks. When a user attempts to log on to a system, the user provides an ID and

	a password The operating system uses the ID to index into the password file and retrieve the plaintext salt and the encrypted password. The salt and user-supplied password are used as input to the encryption routine. If the result matches the stored value, the password is accepted. There are two threats to this password scheme. First, a user can gain access on a machine using a guest account or by some other means and then run a password guessing program, called a password cracker, on that machine. In addition, if an opponent is able to obtain a copy of the password file, then a cracker program can be run on another machine at leisure. This enables the opponent to run through millions of possible passwords in a reasonable period.		
VII (a)	**Distributed Host-Based IDS**  • **Host agent module:** An audit collection module operating as a background process on a monitored system. Its purpose is to collect data on security-related events on the host and transmit these to the central manager. • • **LAN monitor agent module:** Operates in the same fashion as a host agent module except that it analyzes LAN traffic and reports the results to the central manager. • • **Central manager module:** Receives reports from LAN monitor and host agents and processes and correlates these reports to detect intrusion.	Fig—4 Expn--4	8



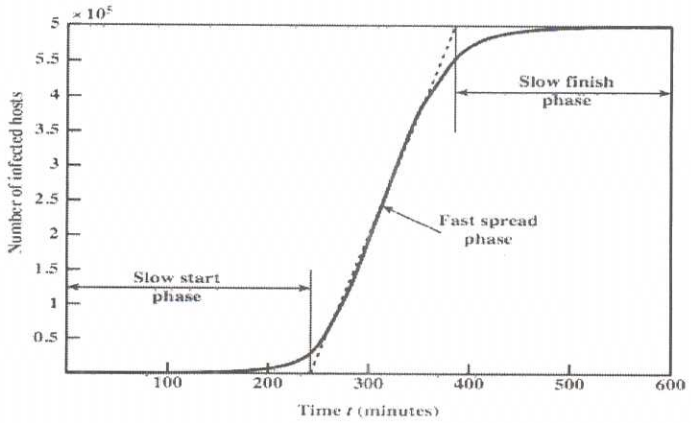
The agent captures each audit record produced by the native audit collection system. A filter is applied that retains only those records that are of security interest. These records are then reformatted into a standardized format referred to as the host audit record (HAR). Next, a template-driven logic module analyzes the records for suspicious activity. At the lowest level, the agent scans for notable events that are of interest independent of any past events. Examples include failed file accesses, accessing system files, and changing a file's access control. At the next higher level, the agent looks for sequences of events, such as known attack patterns (signatures). Finally, the agent looks for anomalous behavior of an individual user based on a historical profile of that user, such as number of programs executed, number of files accessed, and the like. When suspicious activity is detected, an alert is sent to the central manager. The central manager includes an expert system that can draw inferences from received data. The manager may also query individual systems for copies of HARs to correlate with those from other agents.

The LAN monitor agent also supplies information to the central manager. The LAN monitor agent audits host-host connections, services used, and volume of traffic. It searches for significant events, such as sudden changes in network load, the use of security-related services, and network activities such as *rlogin*.

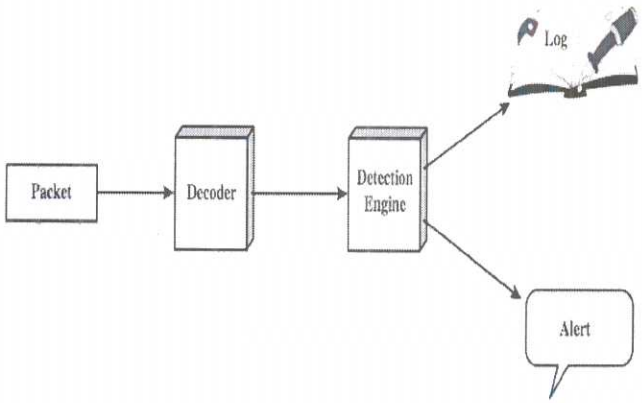
The architecture depicted in Figures 6.2 and 6.3 is quite general and flexible. It offers a foundation for a machine-independent approach that can expand from stand-alone intrusion detection to a system that is able to correlate activity from a number of sites and networks to detect suspicious activity that would otherwise remain undetected.

VII (b)		**Virus Structure** A computer virus has three parts • Infection mechanism:The means by which a virus spreads, enabling it to replicate. The mechanism is also referred to as the infection vector. • Trigger: event or condition determining when the payload is activated or delivered. • Payload: What the virus does, besides spreading. The payload may involve damage or may involve benign but noticeable activity. A virus can be prepended or postpended to an executable program, or it can be embedded in some other fashion. The key to its operation is that the infected program, when invoked, will first execute the virus code and then execute the original code of the program. Once a virus has gained entry to a system by infecting a single program, it is in a position to infect some or all other executable files on that system when the infected program executes. Thus, viral infection can be completely prevented by preventing the virus from gaining entry in the first place. Unfortunately, prevention is extraordinarily difficult because a virus can be part of any program outside a system. Thus, unless one is content to take an absolutely bare piece of iron and write all one's own system and application programs, one is vulnerable. The lack of access controls on early PCs is a key reason why traditional machine code based viruses spread rapidly on these systems. In contrast, while it is easy enough to write a machine code virus for UNIX systems, they were almost never seen in practice due to the existence of access controls on these systems prevented effective propagation of the virus.		

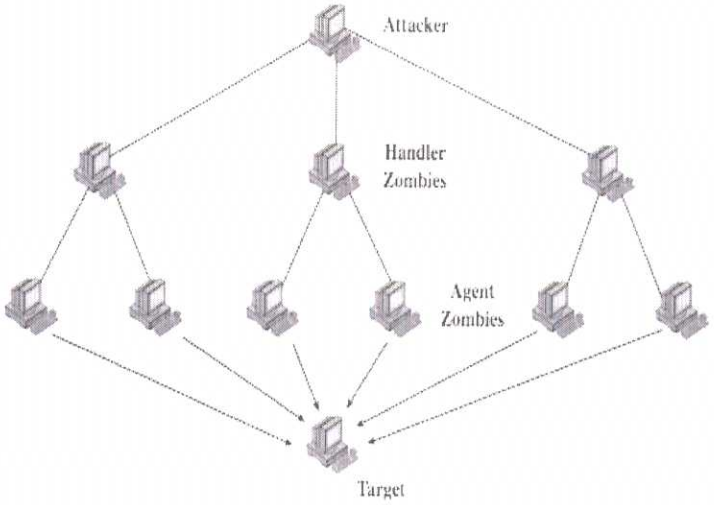
VIII (a)	Worms • replicating program that propagates over net – using email, remote exec, remote login • has phases like a virus: – dormant, propagation, triggering, execution – propagation phase: searches for other systems, connects to it, copies self to it and runs A worm is a program that can replicate itself and send copies from computer to computer across network connections. Upon arrival, the worm may be activated to replicate and propagate again. In addition to propagation, the worm usually performs some unwanted function. Network worm programs use network connections to spread from system to system. Once active within a system, a network worm can behave as a computer virus or bacteria, or it could implant Trojan horse programs or perform any number of disruptive or destructive actions. To replicate itself, a network worm uses some sort of network vehicle such as email, remote execution or remote login capabilities. The new copy of the worm program is then run on the remote system where, in addition to any functions that it performs at that system, it continues to spread in the same fashion. A network worm exhibits the same characteristics as a computer virus: a dormant phase, a propagation phase, a triggering phase, and an execution phase. The propagation phase generally: searches for other systems to infect by examining host tables or similar repositories of remote system addresses; establishes a connection with a remote system; and copies itself to the remote system and cause the copy to be run Worm Propagation Model	Phase —4 Propo gation model --4	8

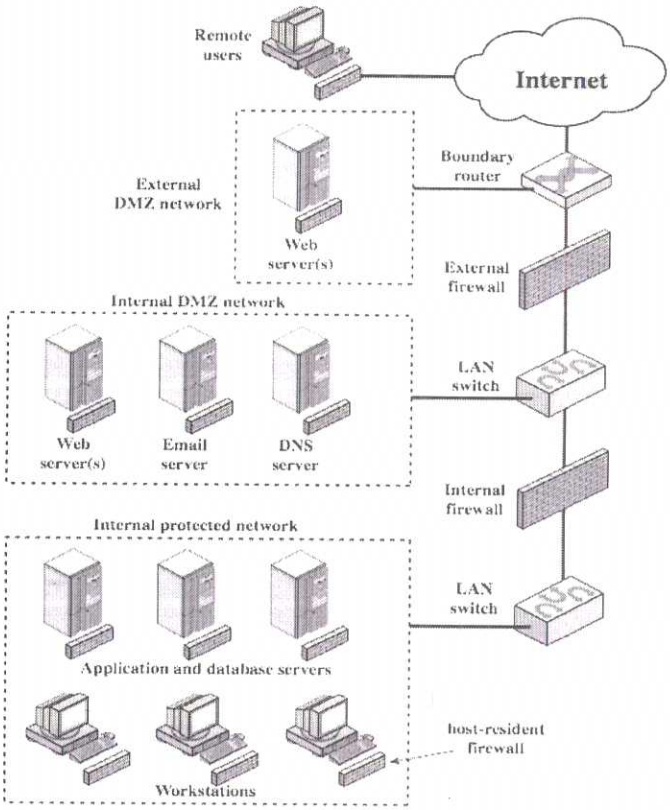


The speed of propagation and the total number of hosts infected depend on a number of factors, including the mode of propagation, the vulnerability or vulnerabilities exploited, and the degree of similarity to preceding attacks. For the latter factor, a an attack that is a variation on a recent previous attack may. In the initial phase, the number of hosts increases exponentially. To see that this is so, consider a simplified case in which a worm is launched from a single host and infects two nearby hosts. Each of these hosts infects two more hosts, and so on. This results in exponential growth. After a time, infecting hosts waste some time attacking already-infected hosts, which reduces the rate of infection. During this middle phase, growth is approximately linear, but the rate of infection is rapid. When most vulnerable computers have been infected, the attack enters a slow finish phase as the worm seeks out those remaining hosts that are difficult to identify. Clearly, the objective in countering a worm is to catch the worm in its slow start phase, at a time when few hosts have been infected. be countered more effectively than a more novel attack

VIII (b)		SNORT Snort is an open source, highly configurable and portable host-based or network-based IDS. Snort is referred to as a lightweight IDS. Snort can perform real-time packet capture, protocol analysis, and content searching and matching. Snort can detect a variety of attacks and probes, based on a set of rules configured by a system administrator. A Snort implementation can be configured as a passive sensor, which monitors traffic but is not in the main transmission path of the traffic, or an inline sensor. In the latter case, Snort can perform intrusion prevention as well as intrusion detection. A Snort installation consists of four logical components, shown here in Figure 6.7: • **Packet Decoder:** efficiently processes each captured packet to identify and isolate protocol headers at the data link, network, transport, and application layers. • **Detection Engine:** does actual work of intrusion detection, analyzing each packet using rules defined for this configuration of Snort by the security administrator. • **Logger:** of each packet that matches a rule, if specified. The security administrator can then use the log file for later analysis. • **Alerter:** can be sent for each detected packet to a file, a UNIX socket, or a database.  Snort is an open source, highly configurable and portable host-based or network-based IDS. Snort is referred to as a lightweight IDS. Snort can perform real-time packet capture, protocol analysis, and content searching and matching. Snort can detect a variety of attacks and probes, based on a set of rules configured by a system administrator. A Snort implementation can be configured as a passive sensor, which monitors traffic but is not in the main transmission path of the traffic, or an inline sensor. In the latter case, Snort can perform intrusion prevention as well as intrusion detection.	Parts —3.5 IDS —3.5	

		A Snort installation consists of four logical components, shown here in Figure 6.7: • **Packet Decoder:** efficiently processes each captured packet to identify and isolate protocol headers at the data link, network, transport, and application layers. • **Detection Engine:** does actual work of intrusion detection, analyzing each packet using rules defined for this configuration of Snort by the security administrator. • **Logger:** of each packet that matches a rule, if specified. The security administrator can then use the log file for later analysis. • **Alerter:** can be sent for each detected packet to a file, a UNIX socket, or a database. SNORT Rules Snort uses a simple, flexible rule definition language that generates the rules used by the detection engine. Although the rules are simple and straightforward to write, they are powerful enough to detect a wide variety of hostile or suspicious traffic. Each rule consists of a fixed header and zero or more options. The header includes: • **Action:** tells Snort what to do when it finds a packet that matches the rule criteria (alert, log, pass, activate, dynamic, drop, reject, sdrop) • **Protocol:** if packet protocol matches this field then analysis proceeds • **Source IP address:** source of packet • **Source port:** for the specified protocol (e.g.,a TCP port). • **Direction:** unidirectional (->) or bidirectional (<->). • **Destination IP address:** destination of packet. • **Destination port**		
IX (a)		Distributed Denial of Service Attacks • have limited volume if single source used • multiple systems allow much higher traffic volumes to form a Distributed Denial of Service (DDoS) Attack • often compromised PC's / workstations – zombies with backdoor programs installed – forming a botnet • e.g. Tribe Flood Network (TFN), TFN2K	Fig—4 Expn--4	8

	 A small number of systems act as handlers controlling a much larger number of agent systems, as shown in Figure 8.4. There are a number of advantages to this arrangement. The attacker can send a single command to a handler, which then automatically forwards it to all the agents under its control. Automated infection tools can also be used to scan for and compromise suitable zombie systems. Once the agent software is uploaded to a newly compromised system, it can contact one or more handlers to automatically notify them of its availability. By this means, the attacker can automatically grow suitable "botnets". TFN and TFN2K use a version of this two-layer command hierarchy. The best defense against being an unwitting participant in a DDoS attack is to prevent your systems from being compromised. For the target of a DDoS attack, the response is the same as for any flooding attack, but with greater volume and complexity.		
IX (b)	Distributed Firewalls	Fog—3.5 Expn-3.5	7

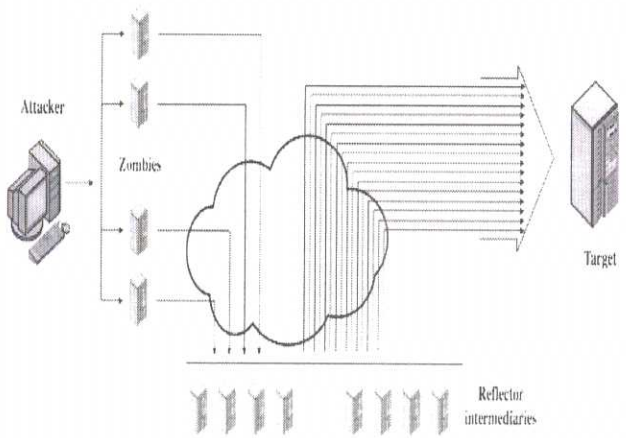


A distributed firewall configuration involves standalone firewall devices plus host-based firewalls working together under a central administrative control. Figure 9.5from the text suggests a distributed firewall configuration. Administrators can configure host-resident firewalls on hundreds of servers and workstation as well as configuring personal firewalls on local and remote user systems. Tools let the network administrator set policies and monitor security across the entire network. These firewalls protect against internal attacks and provide protection tailored to specific machines and applications. Standalone firewalls provide global protection, including internal firewalls and an external firewall, as discussed previously.

With distributed firewalls, it may make sense to establish both an internal and an external DMZ. Web servers that need less protection because they have less critical information on them could be placed in an external DMZ, outside the external firewall. What protection is needed is provided by host-based firewalls on these servers.

An important aspect of a distributed firewall configuration is security monitoring. Such monitoring typically includes log aggregation and analysis, firewall statistics, and fine-grained remote monitoring of individual hosts if needed.

X(a)	**Packet Filtering Firewall** • applies rules to packets in/out of firewall • based on information in packet header – src/dest IP addr & port, IP protocol, interface • typically a list of rules of matches on fields – if match rule says if forward or discard packet • two default policies: – discard - prohibit unless expressly permitted • more conservative, controlled, visible to users – forward - permit unless expressly prohibited • easier to manage/use but less secure Some weaknesses of packet filter firewalls: • Because packet filter firewalls do not examine upper-layer data, they cannot prevent attacks that employ application-specific vulnerabilities or functions. • Because of the limited info available to the firewall, the logging functionality present in packet filter firewalls the same as used to make access control decisions • Most packet filter firewalls do not support advanced user authentication schemes. • They are generally vulnerable to attacks and exploits that take advantage of problems within the TCP/IP specification and protocol stack, such as *network layer address spoofing*. • Finally, due to the small number of variables used in access control decisions, packet filter firewalls are susceptible to security breaches caused by improper configurations. Some of the attacks/countermeasures on packet filtering firewalls are: • **IP address spoofing:** The intruder transmits packets from the outside with a source IP address field containing an address of an internal(assumed trusted) host. The countermeasure is to discard external packets with an inside source address • **Source routing attacks:** specifies the route that a packet should take as it crosses the Internet. The countermeasure is to discard all packets that use this option. • **Tiny fragment attacks:** intruder uses the IP fragmentation option to create extremely small fragments and force the TCP header information into a separate packet fragment, to circumvent filtering rule. It can be defeated by requiring that the first fragment contain most of the transport header. **Stateful Inspection Firewall** • reviews packet header information but also keeps info on TCP connections – typically have low, "known" port no for server – and high, dynamically assigned client port no – simple packet filter must allow all return high port	Packe t filter —4 Statef ul--4	8

	numbered packets back in – stateful inspection packet firewall tightens rules for TCP traffic using a directory of TCP connections – only allow incoming traffic to high-numbered ports for packets matching an entry in this directory – may also track TCP seq numbers as well		
X(b)	Amplification Attacks  Amplification attacks are a variant of reflector attacks, that differ in generating multiple response packets for each original packet sent. This can be achieved by directing the original request to the broadcast address for some network. As a result, all hosts on that network can potentially respond to the request, generating a flood of responses as shown in Figure 8.6. It is only necessary to use a service handled by large numbers of hosts on the intermediate network. A ping flood using ICMP echo request packets was a common choice, used by the "smurf" DoS program. Another possibility is to use a suitable UDP service, such as the echo service, which the "fraggle" DoS program used. The best additional defense against this form of attack is to not allow "directed broadcasts" to be routed into a network from outside.	Fig-3.5 Expn-3.5	7