Unit-3

**Q.1. What are the various components of PKI?**

**Ans:**

Functional elements of a public key infrastructure include certification authorities, registration authorities, repositories, and archives. The users of the PKI come in two flavors: certificate holders and relying parties. An attribute authority is an optional component.

A **certification authority (CA)** is similar to a notary. The CA confirms the identities of parties sending and receiving electronic payments or other communications. Authentication is a necessary element of many formal communications between parties, including payment transactions. In most check-cashing transactions, a driver’s license with a picture is sufficient authentication. A personal identification number (PIN) provides electronic authentication for transactions at a bank automated teller machine (ATM).

A **registration authority (RA)** is an entity that is trusted by the CA to register or vouch for the identity of users to a CA.

A **repository** is a database of active digital certificates for a CA system. The main business of the repository is to provide data that allows users to confirm the status of digital certificates for individuals and businesses that receive digitally signed messages. These message recipients are called relying parties. CAs post certificates and CRLs to repositories.

An **archive** is a database of information to be used in settling future disputes. The business of the archive is to store and protect sufficient information to determine if a digital signature on an “old” document should be trusted.

**1.Certification Authorities:**

* The certification authority, or CA, is the basic building block of the PKI. The CA is a collection of computer hardware, software, and the people who operate it.
* The CA is known by two attributes: its name and its public key. The CA performs four basic PKI functions: issues certificates (i.e., creates and signs them); maintains certificate status information and issues CRLs; publishes its current (e.g., unexpired) certificates and CRLs, so users can obtain the information they need to implement security services; and maintains archives of status information about the expired certificates that it issued.
* These requirements may be difficult to satisfy simultaneously. To fulfill these requirements, the CA may delegate certain functions to the other components of the infrastructure.
* A CA may issue certificates to users, to other CAs, or both. When a CA issues a certificate, it is asserting that the subject (the entity named in the certificate) has the private key that corresponds to the public key contained in the certificate.
* If the CA includes additional information in the certificate, the CA is asserting that information corresponds to the subject as well.
* This additional information might be contact information (e.g., an electronic mail address), or policy information (e.g., the types of applications that can be performed with this public key.)
* When the subject of the certificate is another CA, the issuer is asserting that the certificates issued by the other CA are trustworthy.

**2.Registration Authorities:**

* An RA is designed to verify certificate contents for the CA. Certificate contents may reflect information presented by the entity requesting the certificate, such as a drivers license or recent pay stub. They may also reflect information provided by a third party.
* For example, the credit limit assigned to a credit card reflects information obtained from credit bureaus.
* A certificate may reflect data from the company’s Human Resources department, or a letter from a designated company official.
* For example, Bob’s certificate could indicate that he has signature authority for small contracts. The RA aggregates these inputs and provides this information to the CA.
* Like the CA, the RA is a collection of computer hardware, software, and the person or people who operate it.

**3.PKI Repositories:**

* PKI applications are heavily dependent on an underlying directory service for the distribution of certificates and certificate status information.
* The directory provides a means of storing and distributing certificates, and managing updates to certificates.
* Directory servers are typically implementations of the X.500 standard or subset of this standard. X.500 consists of a series of recommendations and the specification itself references several ISO standards.
* It was designed for directory services that could work across system, corporate, and international boundaries.
* A suite of protocols is specified for operations such as chaining, shadowing, and referral for server-to-server communication, and the Directory Access Protocol (DAP) for client to server communication.
* The Lightweight Directory Access Protocol (LDAP) was later developed as an alternative to DAP. Most directory servers and clients support LDAP, though not all support DAP.

**4.Archives:**

* An archive accepts the responsibility for long term storage of archival information on behalf of the CA. An archive asserts that the information was good at the time it was received, and has not been modified while in the archive.
* The information provided by the CA to the archive must be sufficient to determine if a certificate was actually issued by the CA as specified in the certificate, and valid at that time.
* The archive protects that information through technical mechanisms and appropriate procedures while in its care.
* If a dispute arises at a later date, the information can be used to verify that the private key associated with the certificate was used to sign a document. This permits the verification of signatures on old documents (such as wills) at a later date.

**5.PKI users:**

* PKI Users are organizations or individuals that use the PKI, but do not issue certificates. They rely on the other components of the PKI to obtain certificates, and to verify the certificates of other entities that they do business with.
* End entities include the relying party, who relies on the certificate to know, with certainty, the public key of another entity; and the certificate holder, that is issued a certificate and can sign digital documents.

**Q.2. Explain mesh and hierarchical PKI structure.**

**Ans:**

**PKI ARCHITECTURES:**

* Certificate holders will obtain their certificates from different CAs, depending upon the organization or community in which they are a member. A PKI is typically composed of many CAs linked by trust paths.
* A trust path links a relying party with one or more trusted third parties, such that the relying party can have confidence in the validity of the certificate in use.
* Recipients of a signed message who have no relationship with the CA that issued the certificate for the sender of the message can still validate the sender’s certificate by finding a path between their CA and the one that issued the sender’s certificate.
* The initial challenge is deploying a PKI that can be used throughout an enterprise (e.g., a company or government agency).
* There are two traditional PKI architectures to support this goal, **hierarchical** and **mesh** enterprise architectures.

These three architectures are described below.

1.Enterprise PKI Architecture.

2.Bridge PKI Architecture.

3.Physical Architecture.

* **Enterprise PKI Architectures:**

CAs may be linked in a number of ways. Most enterprises that deploy a PKI will choose either a “**mesh**” or a “**hierarchical**” architecture:

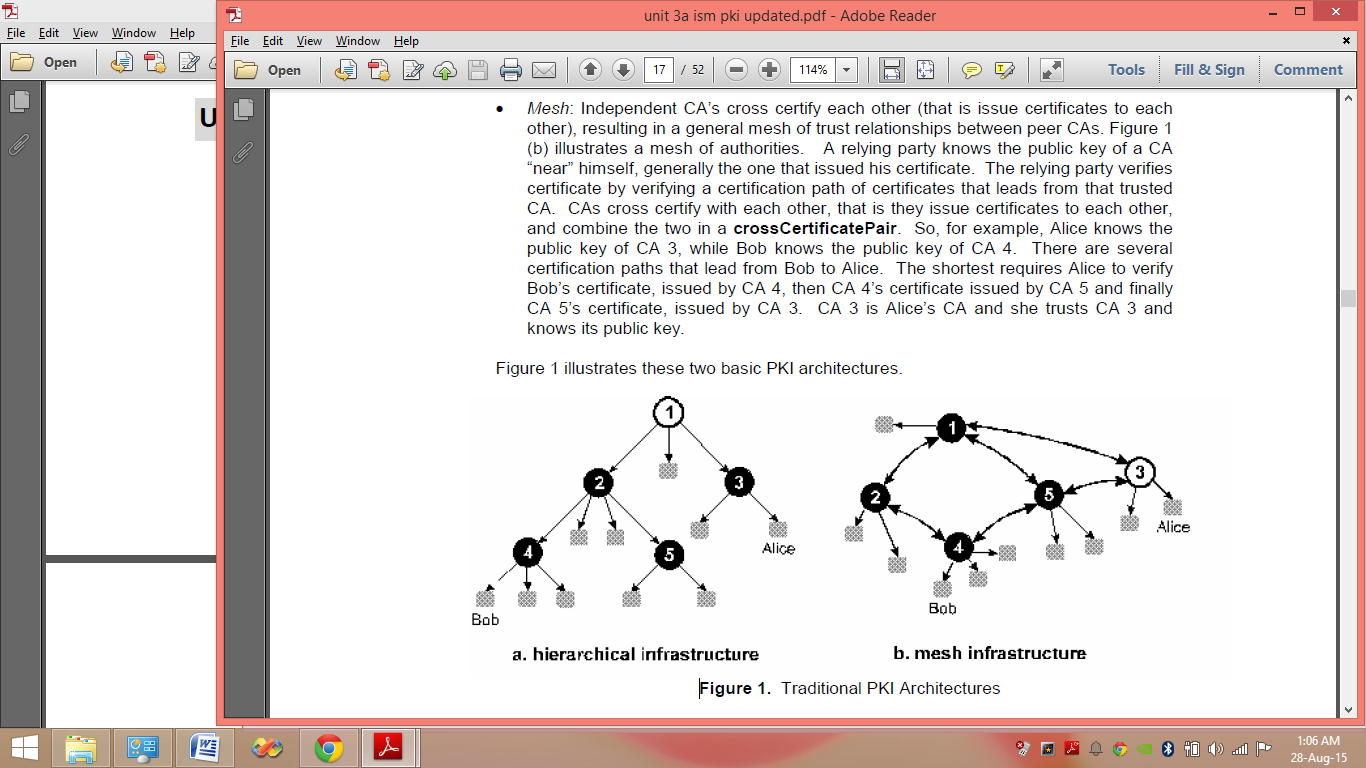
• **Hierarchical***:*

* Authorities are arranged hierarchically under a “root” CA that issues certificates to subordinate CAs.
* These CAs may issue certificates to CAs below them in the hierarchy, or to users. In a hierarchical PKI, every relying party knows the public key of the root CA.
* Any certificate may be verified by verifying the certification path of certificates from the root CA.
* Alice verifies Bob’s certificate, issued by CA 4, then CA 4’s certificate, issued by CA 2, and then CA 2’s certificate issued by CA 1, the root, whose public key she knows.

• **Mesh:**

* Independent CA’s cross certify each other (that is issue certificates to each other), resulting in a general mesh of trust relationships between peer CAs. Figure 1 (b) illustrates a mesh of authorities.
* A relying party knows the public key of a CA “near” himself, generally the one that issued his certificate.
* The relying party verifies certificate by verifying a certification path of certificates that leads from that trusted CA.
* CAs cross certify with each other, that is they issue certificates to each other, and combine the two in a **crossCertificatePair**.
* So, for example, Alice knows the public key of CA 3, while Bob knows the public key of CA 4.
* There are several certification paths that lead from Bob to Alice. The shortest requires Alice to verify Bob’s certificate, issued by CA 4, then CA 4’s certificate issued by CA 5 and finally CA 5’s certificate, issued by CA 3.
* CA 3 is Alice’s CA and she trusts CA 3 and knows its public key.

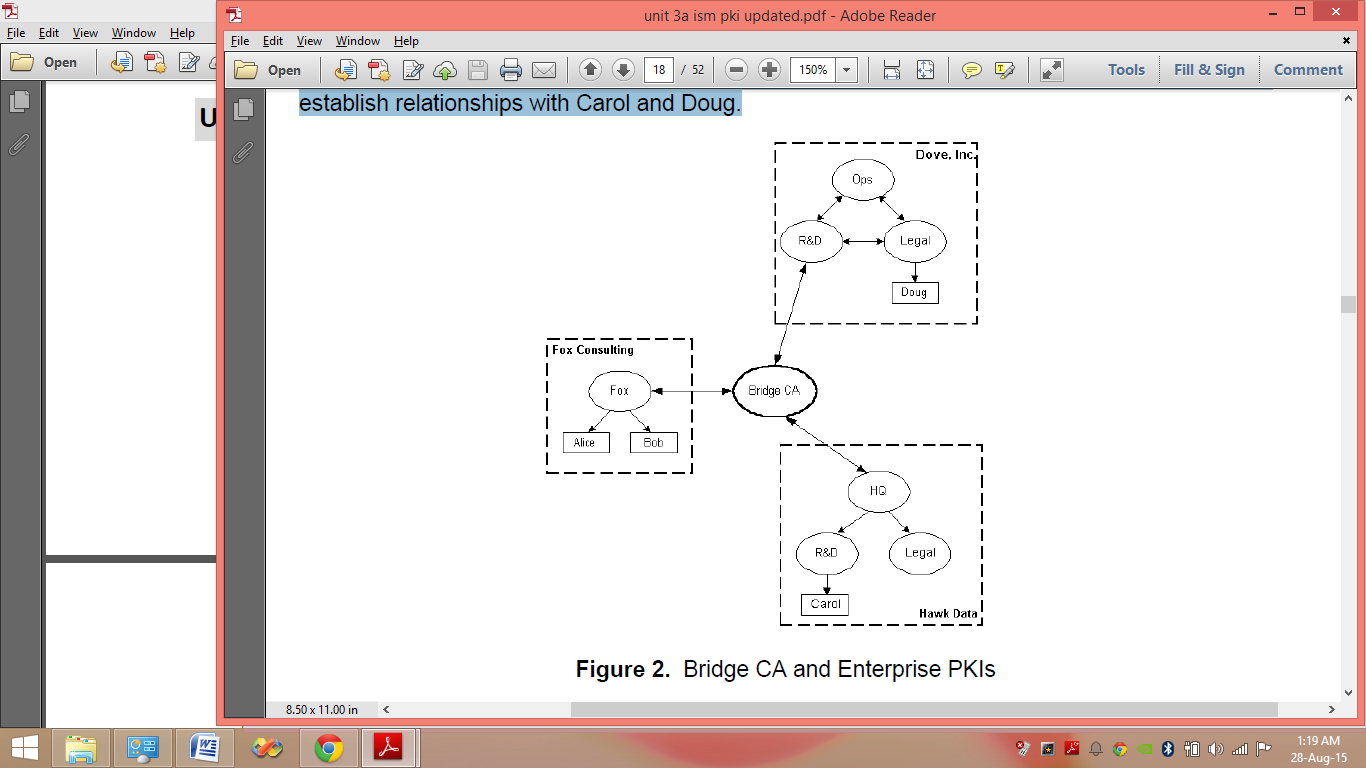
Figure 1 illustrates these two basic PKI architectures.



**Q.3. Explain bridge PKI architecture.**

**Ans:**

* **Bridge PKI Architecture:**
* The Bridge CA architecture was designed to connect enterprise PKIs regardless of the architecture.
* This is accomplished by introducing a new CA, called a Bridge CA, whose sole purpose is to establish relationships with enterprise PKIs.
* Unlike a mesh CA, the Bridge CA does not issue certificates directly to users. Unlike a root CA in a hierarchy, the Bridge CA is not intended for use as a trust point.
* All PKI users consider the Bridge CA an intermediary. The Bridge CA establishes peer-to-peer relationships with different enterprise PKIs.
* These relationships can be combined to form a bridge of trust connecting the users from the different PKIs.
* If the trust domain is implemented as a hierarchical PKI, the Bridge CA will establish a relationship with the root CA.
* If the domain is implemented as a mesh PKI, the bridge will establish a relationship with only one of its CAs. In either case, the CA that enters into a trust relationship with the Bridge is termed a principal CA.
* In Figure 2, the Bridge CA has established relationships with three enterprise PKIs. The first is Bob’s and Alice’s CA, the second is Carol’s hierarchical PKI, and the third is Doug’s mesh PKI.
* None of the users trusts the Bridge CA directly. Alice and Bob trust the CA that issued their certificates; they trust the Bridge CA because the Fox CA issued a certificate to it.
* Carol’s trust point is the root CA of her hierarchy; she trusts the Bridge CA because the root CA issued a certificate to it.
* Doug trusts the CA in the mesh that issued his certificate; he trusts the Bridge CA because there is a valid certification path from the CA that issued him a certificate to the Bridge CA.
* Alice (or Bob) can use the bridge of trust that exists through the Bridge CA to establish relationships with Carol and Doug.



**Q.4. Explain the two basic data structures used in PKIs.**

**Ans:**

**PKI DATA STRUCTURES:**

Two basic data structures are used in PKIs. These are the **public key certificate** and the **certificate revocation lists**. A third data structure, the **attribute certificate**, may be used as an addendum.

**1.X.509 Public Key Certificates:**

The X.509 public key certificate format [IETF 01] has evolved into a flexible and powerful mechanism. It may be used to convey a wide variety of information.

Much of that information is optional, and the contents of mandatory fields may vary as well.

It is important for PKI implementers to understand the choices they face, and their consequences.

Unwise choices may hinder interoperability or prevent support for critical applications.

There are ten common fields: six mandatory and four optional.

The mandatory fields are: the serial number, the certificate signature algorithm identifier, the certificate issuer name, the certificate validity period, the public key, and the subject name. The subject is the party that controls the corresponding private key. There are four optional fields: the version number, two unique identifiers, and the extensions. These optional fields appear only in version 2 and 3 certificates.

**Version**. The version field describes the syntax of the certificate. When the version field is omitted, the certificate is encoded in the original, version 1, syntax.

**Serial number**. The serial number is an integer assigned by the certificate issuer to each certificate.

**Signature**. The signature field indicates which digital signature algorithm (e.g., DSA with SHA-1 or RSA with MD5) was used to protect the certificate.

**Issuer**. The issuer field contains the X.500 distinguished name of the TTP that generated the certificate.

**Validity**. The validity field indicates the dates on which the certificate becomes valid and the date on which the certificate expires.

**Subject.** The subject field contains the distinguished name of the holder of the private key corresponding to the public key in this certificate. The subject may be a CA, a RA, or an end entity.

**Subject public key information.** The subject public key information field contains the

subject’s public key, optional parameters, and algorithm identifier.

**Issuer unique ID** and **subject unique ID**. These fields contain identifiers, and only appear in version 2 or version 3 certificates. The subject and issuer unique identifiers are intended to handle the reuse of subject names or issuer names over time.

**Extensions**. This optional field only appears in version 3 certificates. If present, this field contains one or more certificate extensions. Each extension includes an extension identifier, a criticality flag, and an extension value.

**Subject type**. This field indicates whether a subject is a CA or an end entity.

**Names and identity information**. This field aids in resolving questions about a user’s identity, e.g., are “alice@gsa.gov” and “c=US; o=U.S. Government; ou=GSA; cn=Alice Adams” the same person?

**Key attributes**. This field specifies relevant attributes of public keys, e.g., whether it can be used for key transport, or be used to verify a digital signature.

**Policy information**. This field helps users determine if another user’s certificate can be trusted, whether it is appropriate for large transactions, and other conditions that vary with organizational policies.

**Certificate extensions.** This fieldallow the CA to include information not supported by the basic certificate content. Any organization may define a private extension to meet its particular business requirements.

**2.Certificate Revocation Lists (CRLs):**

* CRLs are the PKI analog of the credit card hot list that store clerks review before accepting large credit card transactions.
* The CRL is protected by a digital signature of the CRL issuer. If the signature can be verified, CRL users know the contents have not been tampered with since the signature was generated. CRLs contain a set of common fields, and may also include an optional set of extensions.

The CRL contains the following fields:

**Version**. The optional version field describes the syntax of the CRL. (In general, the version will be two.)

**Signature**. The signature field contains the algorithm identifier for the digital signature algorithm used by the CRL issuer to sign the CRL.

**Issuer**. The issuer field contains the X.500 distinguished name of the CRL issuer.

**This update**. The this-update field indicates the issue date of this CRL.

**Next update**. The next-update field indicates the date by which the next CRL will be issued.

**Revoked certificates**. The revoked certificates structure lists the revoked certificates. The entry for each revoked certificate contains the certificate serial number, time of revocation, and optional CRL entry extensions.

**CRL Extensions**. The CRL extensions field is used to provide additional information about the whole CRL. Again, this field may only appear if the version is v2.

**3. Attribute Certificates:**

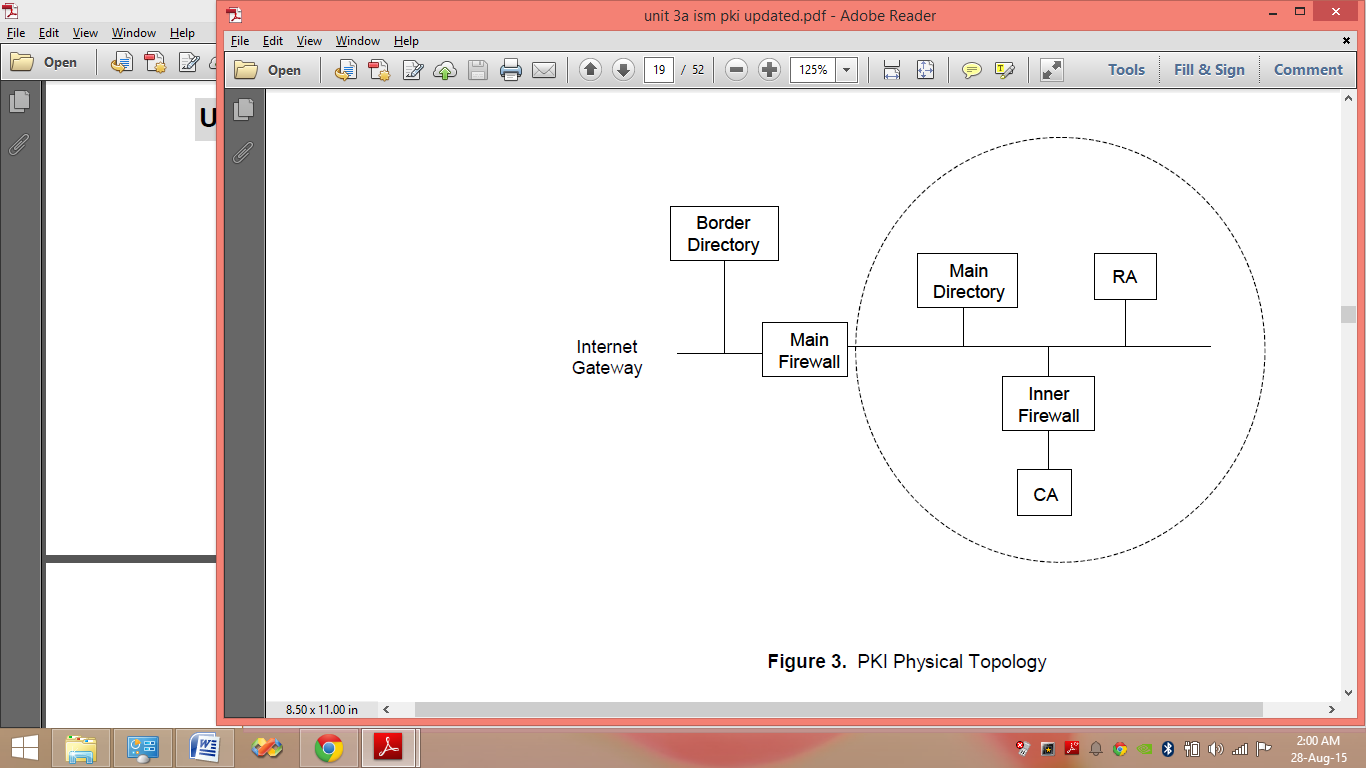
* The public key certificates described in 3.1.1 are focused on the binding between the subject and the public key. The relationship between the subject and public key is expected to be a long-lived relationship.
* Most end entity certificates include a validity period of a year or two years. Organizations seek improved access control. Public key certificates can be used to authenticate the identity of a user, and this identity can be used as an input to access control decision functions.
* However, in many contexts, the identity is not the criterion used for access control decisions. The access control decision may depend upon role, security clearance, group membership, or ability to pay.
* Authorization information, such as membership in a group, often has a shorter lifetime than the binding of the identity and the public key.
* Authorization information could be placed in a public key certificate extension. However, this is not a good strategy for two reasons.
* First, the certificate is likely to be revoked because the authorization information needs to be updated. Revoking and reissuing the public key certificate with updated authorization information is quite expensive.
* Second, the CA that issues public key certificates is not likely to be authoritative for the authorization information. This results in additional steps for the CA to contact the authoritative authorization information source.
* The X.509 attribute certificate (AC) binds attributes to an AC **holder** [X509 97]. This definition is being profiled for use in Internet applications.
* Since the AC does not contain a public key, the AC is used in conjunction with a public key certificate. An access control function may make use of the attributes in an AC, but it is not a replacement for authentication.
* The public key certificate must first be used to perform authentication, then the AC is used to associate attributes with the authenticated identity.

**Q.5. Write a note on physical architecture of PKI.**

**Ans:**

**Physical Architecture:**

* There are numerous ways in which a PKI can be designed physically. It is highly recommended that the major PKI components be implemented on separate systems, that is, the CA on one system, the RA on a different system, and directory servers on other systems.
* Because the systems contain sensitive data, they should be located behind an organization's Internet firewall.
* The CA system is especially important because a compromise to that system could potentially disrupt the entire operations of the PKI and necessitate starting over with new certificates.
* Consequently, placing the CA system behind an additional organizational firewall is recommended so that it is protected both from the Internet and from systems in the organization itself.
* Of course, the organizational firewall would permit communications between the CA and the RA as well as other appropriate systems.
* If distinct organizations wish to access certificates from each other, their directories will need to be made available to each other and possibly to other organizations on the Internet.
* However, some organizations will use the directory server for much more than simply a repository for certificates. The directory server may contain other data considered sensitive to the organization and thus the directory may be too sensitive to be made publicly available.
* A typical solution would be to create a directory that contains only the public keys or certificates, and to locate this directory at the border of the organization - this directory is referred to as a **border directory**.
* A likely location for the directory would be outside the organization’s firewall or perhaps on a protected DMZ segment of its network so that it is still available to the public but better protected from attack.
* Figure 3 illustrates a typical arrangement of PKI-related systems.



* The main directory server located within the organization's protected network would periodically refresh the border directory with new certificates or updates to the existing certificates.
* Users within the organization would use the main directory server, whereas other systems and organizations would access only the border directory.
* When a user in organization A wishes to send encrypted e-mail to a user in organization B, user A would then retrieve user B's certificate from organization B's border directory, and then use the public key in that certificate to encrypt the e-mail.

Chp-2

**Q.6. List the most commonly logged types of information and their potential benefits.**

**Ans:**

The following lists some of the most commonly logged types of information and the potential benefits of each:

**1.Client requests and server responses:**

* **Client requests and server responses**, which can be very helpful in reconstructing sequences of events and determining their apparent outcome.
* If the application logs successful user authentications, it is usually possible to determine which user made each request.
* Some applications can perform highly detailed logging, such as e-mail servers recording the sender, recipients, subject name, and attachment names for each e-mail; Web servers recording each URL requested and the type of response provided by the server; and business applications recording which financial records were accessed by each user.
* This information can be used to identify or investigate incidents and to monitor application usage for compliance and auditing purposes.

2.**Account information:**

* **Account information** such as successful and failed authentication attempts, account changes (e.g., account creation and deletion, account privilege assignment), and use of privileges.
* In addition to identifying security events such as brute force password guessing and escalation of privileges, it can be used to identify who has used the application and when each person has used it.

3.**Usage information:**

* **Usage information** such as the number of transactions occurring in a certain period (e.g., minute, hour) and the size of transactions (e.g., e-mail message size, file transfer size).
* This can be useful for certain types of security monitoring (e.g., a ten-fold increase in e-mail activity might indicate a new e-mail–borne malware threat; an unusually large outbound e-mail message might indicate inappropriate release of information).

4.**Significant operational actions:**

* **Significant operational actions** such as application startup and shutdown, application failures, and major application configuration changes. This can be used to identify security compromises and operational failures.
* Much of this information, particularly for applications that are not used through unencrypted network communications, can only be logged by the applications, which makes application logs particularly valuable for application-related security incidents, auditing, and compliance efforts.
* However, these logs are often in proprietary formats that make them more difficult to use, and the data they contain is often highly context-dependent, necessitating more resources to review their contents.

**Q.7. State & explain the common log management infrastructure functions.**

**Ans:**

* **Log Management Infrastructure:**
* A log management infrastructure consists of the hardware, software, networks, and media used to generate, transmit, store, analyze, and dispose of log data.Most organizations have one or more log management infrastructures.
* **Functions:**
* Log Management infrastructures typically perform several functions that assist in the storage, analysis, and disposal of log data. These functions are normally performed in such a way that they do not alter the original logs.

The following items describe common log management infrastructure functions:

**1.General:**

– **Log parsing** is extracting data from a log so that the parsed values can be used as input for another logging process. A simple example of parsing is reading a text-based log file that contains 10 comma-separated values per line and extracting the 10 values from each line.

– **Event filtering** is the suppression of log entries from analysis, reporting, or long-term storage because their characteristics indicate that they are unlikely to contain information of interest. For example, duplicate entries and standard informational entries might be filtered because they do not provide useful information to log analysts.

– **In event aggregation**, similar entries are consolidated into a single entry containing a count of the number of occurrences of the event. For example, a thousand entries that each record part of a scan could be aggregated into a single entry that indicates how many hosts were scanned.

**2.Storage:**

– **Log rotation** is closing a log file and opening a new log file when the first file is considered to be complete. Log rotation is typically performed according to a schedule (e.g., hourly, daily, weekly) or when a log file reaches a certain size.

– **Log archival** is retaining logs for an extended period of time, typically on removable media, a storage area network (SAN), or a specialized log archival appliance or server. Logs often need to be preserved to meet legal or regulatory requirements. There are two types of log archival: retention and preservation.

– **Log compression** is storing a log file in a way that reduces the amount of storage space needed for the file without altering the meaning of its contents. Log compression is often performed when logs are rotated or archived.

– **Log reduction** is removing unneeded entries from a log to create a new log that is smaller. A similar process is event reduction, which removes unneeded data fields from all log entries.

– **Log conversion** is parsing a log in one format and storing its entries in a second format. For example, conversion could take data from a log stored in a database and save it in an XML format in a text file.

– In **log normalization**, each log data field is converted to a particular data representation and categorized consistently.

– **Log file integrity checking** involves calculating a message digest for each file and storing the message digest securely to ensure that changes to archived logs are detected. The most commonly used message digest algorithms are MD5 and Secure Hash Algorithm 1 (SHA-1).

3.**Analysis:**

– **Event correlation** is finding relationships between two or more log entries. The most common form of event correlation is rule-based correlation, which matches multiple log entries from a single source or multiple sources based on logged values, such as timestamps, IP addresses, and event types.

– **Log viewing** is displaying log entries in a human-readable format. Most log generators provide some sort of log viewing capability; third-party log viewing utilities are also available. Some log viewers provide filtering and aggregation capabilities.

– **Log reporting** is displaying the results of log analysis. Log reporting is often performed to summarize significant activity over a particular period of time or to record detailed information related to a particular event or series of events.

**4.Disposal:**

– **Log clearing** is removing all entries from a log that precede a certain date and time. Log clearing is often performed to remove old log data that is no longer needed on a system because it is not of importance or it has been archived.

**Q.8. What are the various types of network & host based security software.**

**Ans:**

* Most organizations use several types of network-based and host-based security software to detect malicious activity, protect systems and data, and support incident response efforts.
* Accordingly, security software is a major source of computer security log data. Common types of network-based and host-based security software include the following:

**1.Antimalware Software.**

* The most common form of antimalware software is antivirus software, which typically records all instances of detected malware, file and system disinfection attempts, and file quarantines.
* Additionally, antivirus software might also record when malware scans were performed and when antivirus signature or software updates occurred.
* Antispyware software and other types of antimalware software (e.g., rootkit detectors) are also common sources of security information.

**2.Intrusion Detection and Intrusion Prevention Systems.**

* Intrusion detection and intrusion prevention systems record detailed information on suspicious behavior and detected attacks, as well as any actions intrusion prevention systems performed to stop malicious activity in progress.
* Some intrusion detection systems, such as file integrity checking software, run periodically instead of continuously, so they generate log entries in batches instead of on an ongoing basis.

**3.Remote Access Software.**

* Remote access is often granted and secured through virtual private networking (VPN).
* VPN systems typically log successful and failed login attempts, as well as the dates and times each user connected and disconnected, and the amount of data sent and received in each user session.
* VPN systems that support granular access control, such as many Secure Sockets Layer (SSL) VPNs, may log detailed information about the use of resources.

**4.Web Proxies.**

* Web proxies are intermediate hosts through which Web sites are accessed. Web proxies make Web page requests on behalf of users, and they cache copies of retrieved Web pages to make additional accesses to those pages more efficient.
* Web proxies can also be used to restrict Web access and to add a layer of protection between Web clients and Web servers.
* Web proxies often keep a record of all URLs accessed through them.

**5.Vulnerability Management Software.**

* Vulnerability management software, which includes patch management software and vulnerability assessment software, typically logs the patch installation history and vulnerability status of each host, which includes known vulnerabilities and missing software updates.
* Vulnerability management software may also record additional information about hosts’ configurations.
* Vulnerability management software typically runs occasionally, not continuously, and is likely to generate large batches of log entries.

**6.Authentication Servers.**

* Authentication servers, including directory servers and single sign-on servers, typically log each authentication attempt, including its origin, username, success or failure, and date and time.

**7.Routers.**

* Routers may be configured to permit or block certain types of network traffic based on a policy.
* Routers that block traffic are usually configured to log only the most basic characteristics of blocked activity.

**8.Firewalls.**

* Like routers, firewalls permit or block activity based on a policy; however, firewalls use much more sophisticated methods to examine network traffic.Firewalls can also track the state of network traffic and perform content inspection.
* Firewalls tend to have more complex policies and generate more detailed logs of activity than routers.

**9.Network Quarantine Servers.**

* Some organizations check each remote host’s security posture before allowing it to join the network. This is often done through a network quarantine server and agents placed on each host.
* Hosts that do not respond to the server’s checks or that fail the checks are quarantined on a separate virtual local area network (VLAN) segment.
* Network quarantine servers log information about the status of checks, including which hosts were quarantined and for what reasons.

**Q.9.What are the challenges in log management?**

**Ans:**

The most common types of challenges, divided into three groups.

* First, there are several potential problems with the initial generation of logs because of their variety and prevalence.
* Second, the confidentiality, integrity, and availability of generated logs could be breached inadvertently or intentionally.
* Finally, the people responsible for performing log analysis are often inadequately prepared and supported.

### 1.Log Generation and Storage:

In a typical organization, many hosts’ OSs, security software, and other applications generate and store logs. This complicates log management in the following ways:

i.**Many Log Sources.**

* Logs are located on many hosts throughout the organization, necessitating log management to be performed throughout the organization.
* Also, a single log source can generate multiple logs—for example, an application storing authentication attempts in one log and network activity in another log.

ii.**Inconsistent Log Content.**

* Each log source records certain pieces of information in its log entries, such as host IP addresses and usernames.
* For efficiency, log sources often record only the pieces of information that they consider most important.

**iii.Inconsistent Timestamps.**

* Each host that generates logs typically references its internal clock when setting a timestamp for each log entry.
* If a host’s clock is inaccurate, the timestamps in its logs will also be inaccurate. This can make analysis of logs more difficult, particularly when logs from multiple hosts are being analyzed.
* For example, timestamps might indicate that event A happened 45 seconds before event B, when event A actually happened two minutes after event B.

iv.**Inconsistent Log Formats.**

* Many of the log source types use different formats for their logs, such as comma-separated or tab-separated text files,databases, syslog, Simple Network Management Protocol (SNMP), Extensible Markup Language (XML), and binary files.
* Some logs are designed for humans to read, while others are not; some logs use standard formats, while others use proprietary formats.
* Some logs are created not for local storage in a file, but for transmission to another system for processing; a common example of this is SNMP traps.
* For some output formats, particularly text files, there are many possibilities for the sequence of the values in each log entry and the delimiters between the values (e.g., comma-separated values, tab-delimited values, XML).

### 2.Log Protection:

* Because logs contain records of system and network security, they need to be protected from breaches of their confidentiality and integrity.
* For example, logs might intentionally or inadvertently capture sensitive information such as users’ passwords and the content of e-mails.
* This raises security and privacy concerns involving both the individuals that review the logs and others that might be able to access the logs through authorized or unauthorized means.
* Logs that are secured improperly in storage or in transit might also be susceptible to intentional and unintentional alteration and destruction.
* This could cause a variety of impacts, including allowing malicious activities to go unnoticed and manipulating evidence to conceal the identity of a malicious party.
* For example, many rootkits are specifically designed to alter logs to remove any evidence of the rootkits’ installation or execution.

### 3.Log Analysis:

* Within most organizations, network and system administrators have traditionally been responsible for performing log analysis—studying log entries to identify events of interest.
* It has often been treated as a low-priority task by administrators and management because other duties of administrators, such as handling operational problems and resolving security vulnerabilities, necessitate rapid responses.
* Administrators who are responsible for performing log analysis often receive no training on doing it efficiently and effectively, particularly on prioritization.
* Also, administrators often do not receive tools that are effective at automating much of the analysis process, such as scripts and security software tools (e.g., host-based intrusion detection products, security information and event management software).
* Many of these tools are particularly helpful in finding patterns that humans cannot easily see, such as correlating entries from multiple logs that relate to the same event.

**Q.10. Explain log management infrastructure.**

**Ans:**

# Log Management Infrastructure:

* A log management infrastructure consists of the hardware, software, networks, and media used to generate, transmit, store, analyze, and dispose of log data.Most organizations have one or more log management infrastructures.
* **Architecture:**

A log management infrastructure typically comprises the following three tiers:

i.**Log Generation.**

* The first tier contains the hosts that generate the log data. Some hosts run logging client applications or services that make their log data available through networks to log servers in the second tier.
* Other hosts make their logs available through other means, such as allowing the servers to authenticate to them and retrieve copies of the log files.

ii.**Log Analysis and Storage.**

* The second tier is composed of one or more log servers that receive log data or copies of log data from the hosts in the first tier.
* The data is transferred to the servers either in a real-time or near-real-time manner, or in occasional batches based on a schedule or the amount of log data waiting to be transferred.
* Servers that receive log data from multiple log generators are sometimes called collectors or aggregators.
* Log data may be stored on the log servers themselves or on separate database servers.

iii.**Log Monitoring:**

* The third tier contains consoles that may be used to monitor and review log data and the results of automated analysis. Log monitoring consoles can also be used to generate reports.
* In some log management infrastructures, consoles can also be used to provide management for the log servers and clients.
* Also, console user privileges sometimes can be limited to only the necessary functions and data sources for each user.

The second tier—log analysis and storage—can vary greatly in complexity and structure. The simplest arrangement is a single log server that handles all log analysis and storage functions. Examples of more complex second tier arrangements are as follows:

* Multiple log servers that each perform a specialized function, such as one server performing log collection, analysis, and short-term log storage, and another server performing long-term storage.
* Multiple log servers that each perform analysis and/or storage for certain log generators. This can also provide some redundancy. A log generator can switch to a backup log server if its primary log server becomes unavailable. Also, log servers can be configured to share log data with each other, which also supports redundancy.
* Two levels of log servers, with the first level of distributed log servers receiving logs from the log generators and forwarding some or all of the log data they receive to a second level of more centralized log servers. (Additional tiers can be added to this architecture to make it even more flexible, scalable, and redundant.) In some cases, the first level servers act as log caching servers—simply receiving logs from log generators and forwarding them to other log servers. This can be done to protect the second level of log servers from direct attacks, and it is also useful when there are network reliability concerns between the log generators and the second level of log servers, such as those servers being accessible only over the Internet. In that case, having log caching servers on a reliable local network allows the log generators to transfer their logs to those servers, which can then transfer the logs to the second level of log servers when network connectivity permits.

**Q.7. What are the various functions of log management infrastructure?**

**Ans:**

Refer Q.7.

**Q.12. Write short note on Syslog Security**

**Ans:**

### Syslog Security :

* Syslog was developed at a time when the security of logs was not a major consideration. Accordingly, it did not support the use of basic security controls that would preserve the confidentiality, integrity, and availability of logs.
* As the security of logs has become a greater concern, several implementations of syslog have been created that place a greater emphasis on security.
* Most have been based on a proposed standard, RFC 3195, which was designed specifically to improve the security of syslog.
* Implementations based on RFC 3195 can support log confidentiality, integrity, and availability through several features, including the following:

**i.Reliable Log Delivery:**

* Several syslog implementations support the use of Transmission Control Protocol (TCP) in addition to UDP. TCP is a connection-oriented protocol that attempts to ensure the reliable delivery of information across networks.
* Using TCP helps to ensure that log entries reach their destination. Having this reliability requires the use of more network bandwidth; also, it typically takes more time for log entries to reach their destination.
* Some syslog implementations use log caching servers.

**ii.Transmission Confidentiality Protection:**

* RFC 3195 recommends the use of the Transport Layer Security (TLS) protocol to protect the confidentiality of transmitted syslog messages.
* TLS can protect the messages during their entire transit between hosts. TLS can only protect the payloads of packets, not their IP headers, which means that an observer on the network can identify the source and destination of transmitted syslog messages, possibly revealing the IP addresses of the syslog servers and log sources.
* Some syslog implementations use other means to encrypt network traffic, such as passing syslog messages through secure shell (SSH) tunnels.
* Protecting syslog transmissions can require additional network bandwidth and increase the time needed for log entries to reach their destination.

**iii.Transmission Integrity Protection and Authentication:**

RFC 3195 recommends that if integrity protection and authentication are desired, that a message digest algorithm be used. RFC 3195 recommends the use of MD5; proposed revisions to RFC 3195 mention the use of SHA-1.

Some syslog implementations offer additional features that are not based on RFC 3195. The most common extra features are as follows:

**i**. **Robust Filtering:**

* Original syslog implementations allowed messages to be handled differently based on their facility and priority only; no finer-grained filtering was permitted.
* Some current syslog implementations offer more robust filtering capabilities, such as handling messages differently based on the host or program that generated a message, or a regular expression matching content in the body of a message.
* Some implementations also allow multiple filters to be applied to a single message, which provides more complex filtering capabilities.

**ii.Log Analysis:**

* Originally, syslog servers did not perform any analysis of log data; they simply provided a framework for log data to be recorded and transmitted.
* Administrators could use separate add-on programs for analyzing syslog data. Some syslog implementations now have limited log analysis capabilities built in, such as the ability to correlate multiple log entries.

**iii.Event Response.**

Some syslog implementations can initiate actions when certain events are detected. Examples of actions include sending SNMP traps, alerting administrators through pages or e-mails, and launching a separate program or script.

**iv.Alternative Message Formats:**

Some syslog implementations can accept data in non-syslog formats, such as SNMP traps. This can be helpful for getting security event data from hosts that do not support syslog and cannot be modified to do so.

**v**.**Log File Encryption:**

Some syslog implementations can be configured to encrypt rotated log files automatically, protecting their confidentiality. This can also be accomplished through the use of OS or third-party encryption programs.

**vi.Database Storage for Logs.**

Some implementations can store log entries in both traditional syslog files and a database. Having the log entries in a database format can be very helpful for subsequent log analysis.

**vii.Rate Limiting:**

Some implementations can limit the number of syslog messages or TCP connections from a particular source during a certain period of time. This is useful in preventing a denial of service for the syslog server and the loss of syslog messages from other sources.

**Q.13. Explain the Need for Log Management**

**Ans:**

**The Need for Log Management:**

* Log management can benefit an organization in many ways. It helps to ensure that computer security records are stored in sufficient detail for an appropriate period of time.
* Routine log reviews and analysis are beneficial for identifying security incidents, policy violations, fraudulent activity, and operational problems shortly after they have occurred, and for providing information useful for resolving such problems.
* Logs can also be useful for performing auditing and forensic analysis, supporting the organization’s internal investigations, establishing baselines, and identifying operational trends and long-term problems.

The following is a listing of key regulations, standards, and guidelines that help define organizations’ needs for log management:

**i.Federal Information Security Management Act of 2002 (FISMA):**

* FISMA emphasizes the need for each Federal agency to develop, document, and implement an organization-wide program to provide information security for the information systems that support its operations and assets.
* NIST SP 800-53, Recommended Security Controls for Federal Information Systems, was developed in support of FISMA.NIST SP 800-53 is the primary source of recommended security controls for Federal agencies.
* It describes several controls related to log management, including the generation, review, protection, and retention of audit records, as well as the actions to be taken because of audit failure.

**ii.Gramm-Leach-Bliley Act (GLBA):**

* GLBA requires financial institutions to protect their customers’ information against security threats.
* Log management can be helpful in identifying possible security violations and resolving them effectively.

**iii.Health Insurance Portability and Accountability Act of 1996 (HIPAA):**

* HIPAA includes security standards for certain health information. NIST SP 800-66, An Introductory Resource Guide for Implementing the Health Insurance Portability and Accountability Act (HIPAA) Security Rule, lists HIPAA-related log management needs.

**iv**.**Sarbanes-Oxley Act (SOX) of 2002.**

* Although SOX applies primarily to financial and accounting practices, it also encompasses the information technology (IT) functions that support these practices.
* SOX can be supported by reviewing logs regularly to look for signs of security violations, including exploitation, as well as retaining logs and records of log reviews for future review by auditors.

**v.Payment Card Industry Data Security Standard (PCI DSS):**

* PCI DSS applies to organizations that “store, process or transmit cardholder data” for credit cards. One of the requirements of PCI DSS is to “track…all access to network resources and cardholder data”.

**Q.14. List& Explain the classic categories of malware.**

**Ans:**

**Forms of Malware:**

* Malware has become the greatest external threat to most hosts, causing damage and requiring extensive recovery efforts within most organizations. The following are the classic categories of malware:

**1.Viruses:**

* A virus self-replicates by inserting copies of itself into host programs or data files. Viruses are often triggered through user interaction, such as opening a file or running a program. Viruses can be divided into the following two subcategories:

– **Compiled Viruses:**

* A compiled virus is executed by an operating system. Types of compiled viruses include file infector viruses, which attach themselves to executable programs; boot sector viruses, which infect the master boot records of hard drives or the boot sectors of removable media; and multipartite viruses, which combine the characteristics of file infector and boot sector viruses.

– **Interpreted Viruses:**

* Interpreted viruses are executed by an application. Within this subcategory, macro viruses take advantage of the capabilities of applications’ macro programming language to infect application documents and document templates, while scripting viruses infect scripts that are understood by scripting languages processed by services on the OS.

**2.Worms:**

* A worm is a self-replicating, self-contained program that usually executes itself without user intervention. Worms are divided into two categories:

– **Network Service Worms:**

* A network service worm takes advantage of a vulnerability in a network service to propagate itself and infect other hosts.

– **Mass Mailing Worms:**

* A mass mailing worm is similar to an email-borne virus but is self-contained, rather than infecting an existing file.

**3.Trojan Horses:**

* A Trojan horse is a self-contained, non-replicating program that, while appearing to be benign, actually has a hidden malicious purpose.
* Trojan horses either replace existing files with malicious versions or add new malicious files to hosts. They often deliver other attacker tools to hosts.

**4.Malicious Mobile Code:**

* Malicious mobile code is software with malicious intent that is transmitted from a remote host to a local host and then executed on the local host, typically without the user’s explicit instruction.
* Popular languages for malicious mobile code include Java, ActiveX, JavaScript, and VBScript.

**5.Blended Attacks:**

* A blended attack uses multiple infection or transmission methods. For example, a blended attack could combine the propagation methods of viruses and worms.

Chp-3

**Q.15. List& Explain the popular attacker tools.**

**Ans:**

**Attacker Tools:**

Various types of attacker tools might be delivered to a host by malware. These tools allow attackers to have unauthorized access to or use of infected hosts and their data, or to launch additional attacks. Popular types of attacker tools are as follows:

1.**Backdoors:**

* A backdoor is a malicious program that listens for commands on a certain TCP or UDP port.
* Most backdoors allow an attacker to perform a certain set of actions on a host, such as acquiring passwords or executing arbitrary commands.
* Types of backdoors include zombies (better known as bots), which are installed on a host to cause it to attack other hosts, and remote administration tools, which are installed on a host to enable a remote attacker to gain access to the host’s functions and data as needed.

**2.Keystroke Loggers:**

* A keystroke logger monitors and records keyboard use. Some require the attacker to retrieve the data from the host, whereas other loggers actively transfer the data to another host through email, file transfer, or other means.

**3.Rootkits:**

* A rootkit is a collection of files that is installed on a host to alter its standard functionality in a malicious and stealthy way. A rootkit typically makes many changes to a host to hide the rootkit’s existence, making it very difficult to determine that the rootkit is present and to identify what the rootkit has changed.

**4.Web Browser Plug-Ins:**

* A web browser plug-in provides a way for certain types of content to be displayed or executed through a web browser. Malicious web browser plug-ins can monitor all use of a browser.

**5.E-Mail Generators:**

* An email generating program can be used to create and send large quantities of email, such as malware and spam, to other hosts without the user’s permission or knowledge.

**6.Attacker Toolkits:**

* Many attackers use toolkits containing several different types of utilities and scripts that can be used to probe and attack hosts, such as packet sniffers, port scanners, vulnerability scanners, password crackers, and attack programs and scripts.

**Q.16. What are the recommended capabilities of an antivirus software?**

**Ans:**

Antivirus software is the most commonly used technical control for malware threat mitigation. There are many brands of antivirus software, with most providing similar protection through the following recommended capabilities:

* Scanning critical host components such as startup files and boot records.
* Watching real-time activities on hosts to check for suspicious activity; a common example is scanning all email attachments for known malware as emails are sent and received. Antivirus software should be configured to perform real-time scans of each file as it is downloaded, opened, or executed, which is known as *on-access scanning*.
* Monitoring the behavior of common applications, such as email clients, web browsers, and instant messaging software. Antivirus software should monitor activity involving the applications most likely to be used to infect hosts or spread malware to other hosts.
* Scanning files for known malware. Antivirus software on hosts should be configured to scan all hard drives regularly to identify any file system infections and, optionally, depending on organization security needs, to scan removable media inserted into the host before allowing its use. Users should also be able to launch a scan manually as needed, which is known as *on-demand scanning*.
* Identifying common types of malware as well as attacker tools.
* *Disinfecting* files, which refers to removing malware from within a file, and *quarantining* files, which means that files containing malware are stored in isolation for future disinfection or examination. Disinfecting a file is generally preferable to quarantining it because the malware is removed and the original file restored; however, many infected files cannot be disinfected. Accordingly, antivirus software should be configured to attempt to disinfect infected files and to either quarantine or delete files that cannot be disinfected.

**Q.17. Write a note on sandboxing.**

**Ans:**

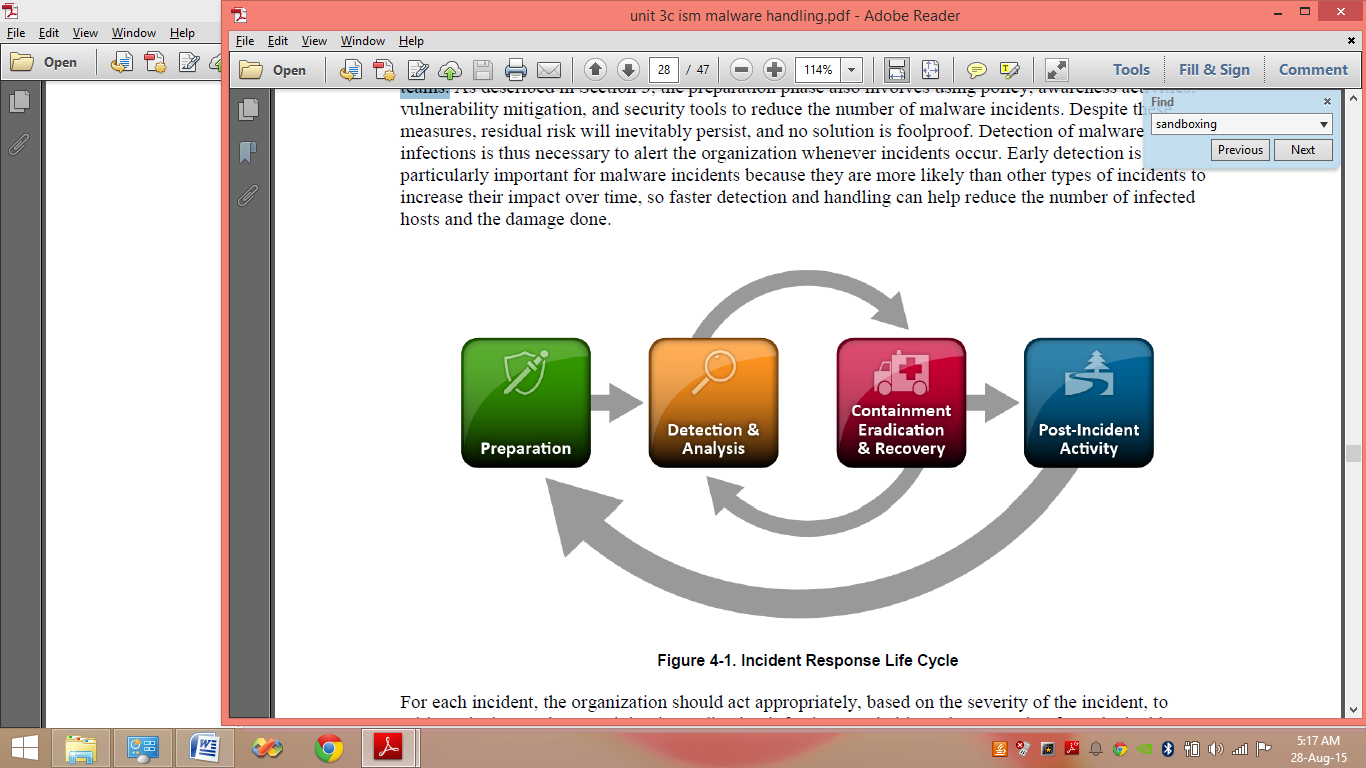
**Sandboxing:**

* *Sandboxing* refers to a security model where applications are run within a sandbox—a controlled environment that restricts what operations the applications can perform and that isolates them from other applications running on the same host.
* In a sandbox security model, typically only authorized “safe” operations may be performed within the sandbox; the sandbox prohibits applications within the sandbox from performing any other operations.
* The sandbox also restricts access to system resources, such as memory and the file system, to keep the sandbox’s applications isolated from the host’s other applications.
* Sandboxing provides several benefits in terms of malware incident prevention and handling. By limiting the operations available, it can prevent malware from performing some or all of the malicious actions it is attempting to execute; this could prevent the malware from succeeding or reduce the damage it causes.
* And the sandboxing environment—the isolation—can further reduce the impact of the malware by restricting what information and functions the malware can access.
* Another benefit of sandboxing is that the sandbox itself can be reset to a known good state every time it is initialized.

**Q.18. Explain malware incident response life cycle in detail.**

**Ans:**

The initial phase of malware incident response involves performing preparatory activities, such as developing malware-specific incident handling procedures and training programs for incident response teams.

****

**1.Preparation:**

* Organizations should perform preparatory measures to ensure that they are capable of responding effectively to malware incidents.
* Preparation phase describe several recommended preparatory measures, including building and maintaining malware-related skills within the incident response team, facilitating communication and coordination throughout the organization, and acquiring necessary tools and resources.

**i**. **Building and Maintaining Malware-Related Skills:**

* Incident handlers should be familiar with the organization’s implementations and configurations of malware detection tools so that they are better able to analyze supporting data and identify the characteristics of threats.
* Incident handlers doing in-depth malware analysis should have strong skills in that area and be familiar with the numerous tools for malware analysis.

**ii**. **Facilitating Communication and Coordination:**

* To improve communication and coordination, an organization should designate in advance a few individuals or a small team to be responsible for coordinating the organization’s responses to malware incidents.
* The coordinator’s primary goal is to maintain situational awareness by gathering all pertinent information, making decisions that are in the best interests of the organization, and communicating pertinent information and decisions to all relevant parties in a timely manner.
* For malware incidents, the relevant parties often include end users, who might be given instructions on how to avoid infecting their hosts, how to recognize the signs of an infection, and what to do if a host appears to be infected.
* The coordinator also needs to provide technical guidance and instructions to all staff assisting with containment, eradication, and recovery efforts, as well as giving management regular updates on the status of the response and the current and likely future impact of the incident.

**iii**. **Acquiring Tools and Resources :**

* Organizations should also ensure that they have the necessary tools (hardware and software) and resources to assist in malware incident handling.

**2.Detection and Analysis:**

* Organizations should strive to detect and validate malware incidents rapidly to minimize the number of infected hosts and the amount of damage the organization sustains.

**i.Identifying Malware Incident Characteristics:**

* As part of the analysis and validation process, incident handlers typically identify characteristics of the malware activity by examining detection sources.
* Understanding the activity’s characteristics is very helpful in assigning an appropriate priority to the incident response efforts and planning effective containment, eradication, and recovery activities.

ii. **Identifying Infected Hosts:**

* Identifying hosts that are infected by malware is part of every malware incident. Once identified, infected hosts can undergo the appropriate containment, eradication, and recovery actions.
* Unfortunately, identifying all infected hosts is often complicated by the dynamic nature of computing.
* The possible categories of infected host identification techniques: forensic, active, and manual.

a.Forensic Identification:

DNS Server Logs.

Other Application Server Logs.

Network Forensic Tools.

Network Device Logs.

b.Active Identification:

Security Automation.

Custom Network-Based IPS or IDS Signature.

Packet Sniffers and Protocol Analyzers.

c.Manual Identification:

d.Identification Recommendations

**iii.Prioritizing Incident Response:**

* Once a malware incident has been validated, the next activity is to prioritize its handling.
* Certain forms of malware, such as worms, tend to spread very quickly and can cause a substantial impact in minutes or hours, so they often necessitate a high-priority response.

**iv**. **Malware Analysis:**

* It is significantly faster and easier to analyze malware by monitoring it during execution.
* Such active approaches are best performed on malware test systems instead of production hosts, to minimize possible damage caused by allowing the malware to execute.

3. **Containment:**

* Containment of malware has two major components: stopping the spread of the malware and preventing further damage to hosts. Nearly every malware incident requires containment actions. In addressing an incident, it is important for an organization to decide which methods of containment to employ initially, early in the response.
* Containment of isolated incidents and incidents involving noninfectious forms of malware is generally straightforward, involving such actions as disconnecting the affected hosts from networks or shutting down the hosts.
* For more widespread malware incidents, such as fast-spreading worms, organizations should use a strategy that contains the incident for most hosts as quickly as possible; this should limit the number of machines that are infected, the amount of damage that is done, and the amount of time that it will take to fully recover all data and services.

**4. Eradication:**

The primary goal of eradication is to remove malware from infected hosts. Because of the potential need for extensive eradication efforts, organizations should be prepared to use various combinations of eradication techniques simultaneously for different situations.

Organizations should also consider performing awareness activities that set expectations for eradication and recovery efforts; these activities can be helpful in reducing the stress that major malware incidents can cause.

**5.Recovery:**

* The two main aspects of recovery from malware incidents are restoring the functionality and data of infected hosts and removing temporary containment measures.
* Additional actions to restore hosts are not necessary for most malware incidents that cause limited host damage (for example, an infection that simply altered a few data files and was completely removable with antivirus software).
* For malware incidents that are far more damaging, such as Trojan horses, rootkits, or backdoors, corrupting thousands of system and data files, or wiping out hard drives, it is often best to first rebuild the host, then secure the host so that it is no longer vulnerable to the malware threat.

**6.Post-Incident Activity:**

* Because the handling of malware incidents can be extremely expensive, it is particularly important for organizations to conduct a robust assessment of lessons learned after major malware incidents to prevent similar incidents from occurring.
* Capturing the lessons learned from the handling of such incidents should help an organization improve its incident handling capability and malware defenses, including identifying needed changes to security policy, software configurations, and malware detection and prevention software deployments.

**Q.19. List and explain the major component of containment of malware.**

**Ans:**

* Containment of malware has two major components:

i.stopping the spread of the malware and

ii.preventing further damage to hosts.

* Nearly every malware incident requires containment actions. In addressing an incident, it is important for an organization to decide which methods of containment to employ initially, early in the response.
* Containment of isolated incidents and incidents involving noninfectious forms of malware is generally straightforward, involving such actions as disconnecting the affected hosts from networks or shutting down the hosts.
* For more widespread malware incidents, such as fast-spreading worms, organizations should use a strategy that contains the incident for most hosts as quickly as possible; this should limit the number of machines that are infected, the amount of damage that is done, and the amount of time that it will take to fully recover all data and services.
* In containing a malware incident, it is also important to understand that stopping the spread of malware does not necessarily prevent further damage to hosts. Malware on a host might continue to exfiltrate sensitive data, replace OS files, or cause other damage.
* In addition, some instances of malware are designed to cause additional damage when network connectivity is lost or other containment measures are performed.
* For example, an infected host might run a malicious process that contacts another host periodically. If that connectivity is lost because the infected host is disconnected from the network, the malware might overwrite all the data on the host’s hard drive.
* For these reasons, handlers should not assume that just because a host has been disconnected from the network, further damage to the host has been prevented, and in many cases, should begin eradication efforts as soon as possible to prevent more damage.
* Organizations should have strategies and procedures in place for making containment-related decisions that reflect the level of risk acceptable to the organization.
* For example, an organization might decide that infected hosts performing critical functions should not be disconnected from networks or shut down if the likely damage to the organization from those functions being unavailable would be greater than the security risks posed by not isolating or shutting down the host.
* Containment strategies should support incident handlers in selecting the appropriate combination of containment methods based on the characteristics of a particular situation.
* Containment methods can be divided into four basic categories: relying on user participation, performing automated detection, temporarily halting services, and blocking certain types of network connectivity.

Chp – 3

**Q.20. Explain the three main categories of patch and vulnerability metrics.**

**Ans:**

There are three main categories of patch and vulnerability metrics: susceptibility to attack, mitigation response time, and cost.

#### 1.Measuring a System’s Susceptibility to Attack :

* An organization’s susceptibility to attack can be approximated by several measurements. An organization can measure the number of patches needed, the number of vulnerabilities, and the number of network services running on a per system basis.
* These measurements should be taken individually for each computer within the system, and the results then aggregated to determine the system-wide result.
* Both raw results and ratios (e.g., number of vulnerabilities per computer) are important.
* The raw results help reveal the overall risk a system faces because the more vulnerabilities, unapplied patches, and exposed network services that exist, the greater the chance that the system will be penetrated.
* Large systems consisting of many computers are thus inherently less secure than smaller similarly configured systems.

**i.Number of Patches:**

* Measuring the number of patches needed per system is natural for organizations that have deployed enterprise patch management tools, since these tools automatically provide such data.
* The number of patches needed is of some value in approximating an organization’s susceptibility to attack, but its effectiveness is limited because a particular security patch may fix one or many vulnerabilities, and these vulnerabilities may be of varying levels of severity.

**ii.Number of Vulnerabilities:**

* Measuring the number of vulnerabilities that exist per system is a better measure of an organization's susceptibility to attack, but still is far from perfect.

**iii.Number of Network Services:**

* The last example of an attack susceptibility metric is measuring the number of network services running per system.The concept behind this metric is that each network service represents a potential set of vulnerabilities, and thus there is an enhanced security risk when systems run additional network services.

#### 2.Mitigation Response Time:

#### It is also important to measure how quickly an organization can identify, classify, and respond to a new vulnerability and mitigate the potential impact within the organization.

#### Response time has become increasingly important, because the average time between a vulnerability announcement and an exploit being released has decreased dramatically in the last few years.

#### There are three primary response time measurements that can be taken: vulnerability and patch identification, patch application, and emergency security configuration changes.

**i.Response Time for Vulnerability and Patch Identification**:

* This metric measures how long it takes the PVG to learn about a new vulnerability or patch.
* Timing should begin from the moment the vulnerability or patch is publicly announced.

**ii.Response Time for Patch Application**:

* This metric measures how long it takes to apply a patch to all relevant IT devices within the system.
* Timing should begin from the moment the PVG becomes aware of a patch.

**iii.Response Time for Emergency Configuration Changes:**

* This metric applies in situations where a vulnerability exists that must be mitigated but where there is no patch.
* In such cases the organization is forced to make emergency configuration changes that may reduce functionality to protect the organization from exploitation of the vulnerability. Such changes are often done at the firewall, e-mail server, Web server, central file server, or servers in the DMZ.

#### 3.Cost :

Measuring the cost of patch and vulnerability management is difficult because the actions are often split between many different personnel and groups.

**i.Cost of the Patch and Vulnerability Group**:

* This measurement is fairly easy to obtain since the PVG personnel are easily identifiable and the percentage of each person’s time dedicated to PVG support should be well-documented.
* When justifying the cost of the PVG to management, it will be useful to estimate the amount of system administrator labor that has been saved by centralizing certain functions within the PVG.

**ii.Cost of System Administrator Support:**

* This measurement is always difficult to take with accuracy but is important nonetheless. The main problem is that, historically, system administrators have not been asked to calculate the amount of time they spend on security, much less on security patch and vulnerability management.
* As organizations improve in their overall efforts to measure the real cost of IT security, measuring the cost of patch and vulnerability measurement with respect to system administrator time will become easier.

**iii.Cost of Enterprise Patch and Vulnerability Management Tools:**

* This measurement includes patching tools, vulnerability scanning tools, vulnerability Web portals, vulnerability databases, and log analysis tools (used for verifying patches).
* It should not include intrusion detection, intrusion prevention, and log analysis tools (used for intrusion detection).
* Estimated annual cost = Sum of annual maintenance for each product + Sum of

(purchase price or upgrade price / life expectancy in

years) for each product

**iv.Cost of Program Failures :**

* This measurement calculates the total cost of the business impact of all incidents that could have been prevented if the patch and vulnerability mitigation program had been more effective, as well as all problems caused by the patching process itself, such as a patch inadvertently breaking an application.

**Q.21.What is The Patch and Vulnerability Group & what are their duties?**

**Ans:**

### The Patch and Vulnerability Group :

* The PVG should be a formal group that incorporates representatives from information security and operations.
* These representatives should include individuals with knowledge of vulnerability and patch management, as well as system administration, intrusion detection, and firewall management.

**The duties of the PVG are outlined below. Subsequent sections discuss certain duties in more detail.**

1. **Create a System Inventory:**

* The PVG should use existing inventories of the organization’s IT resources to determine which hardware equipment, operating systems, and software applications are used within the organization.
* The PVG should also maintain a manual inventory of IT resources not captured in the existing inventories. Section 2.2 contains detailed guidance on creating an inventory.

1. **Monitor for Vulnerabilities, Remediations, and Threats:**

* The PVG is responsible for monitoring security sources for vulnerability announcements, patch and non-patch remediations, and emerging threats that correspond to the software within the PVG’s system inventory.

1. **Prioritize Vulnerability Remediation:**

* The PVG should prioritize the order in which the organization addresses vulnerability remediation.

1. **Create an Organization-Specific Remediation Database:**

* The PVG should create a database of remediations that need to be applied to the organization.

1. **Conduct Generic Testing of Remediations:**

* The PVG should be able to test patches and non-patch remediations on IT devices that use standardized configurations. This will avoid the need for local administrators to perform redundant testing.
* The PVG should also work closely with local administrators to test patches and configuration changes on important systems.

1. **Deploy Vulnerability Remediations:**

* The PVG should oversee vulnerability remediation. Section 2.7 contains information on this process.

1. **Distribute Vulnerability and Remediation Information to Local Administrators:**

* The PVG is responsible for informing local administrators about vulnerabilities and remediations that correspond to software packages included within the PVG scope and that are in the organizational software inventory.

1. **Perform Automated Deployment of Patches:**

* The PVG should deploy patches automatically to IT devices using enterprise patch management tools. Alternately, the PVG could work closely with the group actually running the patch management tools.
* Automated patching tools allow an administrator to update hundreds or even thousands of systems from a single console.
* Deployment is fairly simple when there are homogeneous computing platforms, with standardized desktop systems and similarly configured servers. Multiplatform environments, nonstandard desktop systems, legacy computers, and computers with unusual configurations may also be integrated.

1. **Configure Automatic Update of Applications Whenever Possible and Appropriate.**

* Many newer applications provide a feature that checks the vendor’s Web site for updates. This feature can be very useful in minimizing the level of effort required to identify, distribute, and install patches.
* However, some organizations may not wish to implement this feature because it might interfere with their configuration management process.
* A recommended option would be a locally distributed automated update process, where the patches are made available from the organization’s network. Applications can then be updated from the local network instead of from the Internet.

**10**.**Verify Vulnerability Remediation Through Network and Host Vulnerability Scanning:**

* The PVG should verify that vulnerabilities have been successfully remediated.

**11.Vulnerability Remediation Training:**

* The PVG should train administrators on how to apply vulnerability remediations. In organizations that rely on end users to patch computers, the PVG must also train users on this function.

**Q.22. What are the primary methods of remediation that can be applied to an affected system?**

**Ans:**

* Organizations should deploy vulnerability remediations to all systems that have the vulnerability, even for systems that are not at immediate risk of exploitation.
* Vulnerability remediations should also be incorporated into the organization’s standard builds and configurations for hosts.
* There are three primary methods of remediation that can be applied to an affected system: the installation of a software patch, the adjustment of a configuration setting, and the removal of the affected software.

**1.Security Patch Installation.**

* Applying a security patch (also called a “fix” or “hotfix”) repairs the vulnerability, since patches contain code that modifies the software application to address and eliminate the problem. Patches downloaded from vendor Web sites are typically the most up-to-date and are likely free of malicious code.

**2.Configuration Adjustment.**

* Adjusting how an application or security control is configured can effectively block attack vectorsand reduce the threat of exploitation.
* Common configuration adjustments include disabling services and modifying privileges, as well as changing firewall rules and modifying router access controls.
* Settings of vulnerable software applications can be modified by adjusting file attributes or registry settings.

**3.Software Removal.**

* Removing or uninstalling the affected software or vulnerable service eliminates the vulnerability and any associated threat. This is a practical solution when an application is not needed on a system.
* Determining how the system is used, removing unnecessary software and services, and running only what is essential for the system’s purpose is a recommended security practice.
* The mitigation of vulnerabilities and threats may be as simple as modifying a configuration setting, or as involved as the installation of a completely new version of the software.
* No simple patch application methodology applies to all software and operating systems. Before performing the remediation, the administrator may want to conduct a full backup of the system to be patched.
* This will allow for a timely restoration of the system to previous state if the patch has an unintended or unexpected impact on the host.

**Q.23. Who are involved in log management planning? Explain their responsibilities. ( Chp-2 )**

**Ans:**

As part of the log management planning process, an organization should define the roles and responsibilities of individuals and teams who are expected to be involved in log management.

Some of the organization’s security administrators act as log management infrastructure administrators, with responsibilities such as the following:

􀀟 Contacting system-level administrators to get additional information regarding an event or to request that they investigate a particular event

􀀟 Identifying changes needed to system logging configurations (e.g., which entries and data fields are sent to the centralized log servers, what log format should be used) and informing system-level administrators of the necessary changes

􀀟 Initiating responses to events, including incident handling and operational problems (e.g., a failure of a log management infrastructure component)

􀀟 Ensuring that old log data is archived to removable media and disposed of properly once it is no longer needed

􀀟 Cooperating with requests from legal counsel, auditors, and others

􀀟 Monitoring the status of the log management infrastructure (e.g., failures in logging software or log archival media, failures of local systems to transfer their log data) and initiating appropriate responses when problems occur

􀀟 Testing and implementing upgrades and updates to the log management infrastructure’s components

􀀟 Maintaining the security of the log management infrastructure.

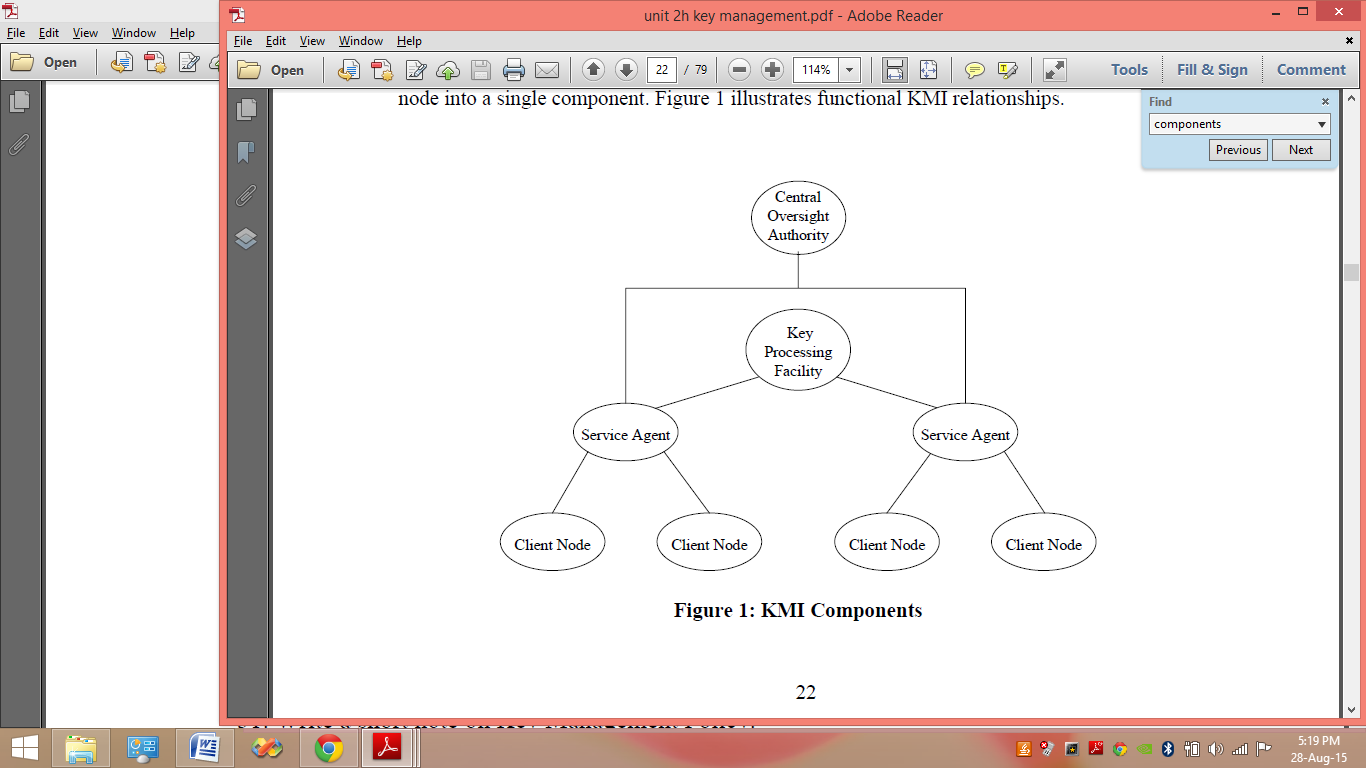
**Q.24. What are the steps included in developing logging policies? ( Chp-2 )**

**Ans:**

**Q.25. List and explain the components of key management infrastructure.(Unit-2)**

**Ans:**

* A KMI is designed to incorporate a set of functional elements, or nodes, that collectively provide unified and seamless protection policy enforcement and key management services.
* Four distinct functional nodes are identified for the generation, distribution, and management of cryptographic keys: a central oversight authority, key processing facility(ies), service agents, and client nodes.
* It should be noted that organizations may choose to combine the functionality of more than one node into a single component. Figure 1 illustrates functional KMI relationships.

****

## **1.Central Oversight Authority:**

## 

The KMI’s central oversight authority is the entity that provides overall KMI data synchronization and system security oversight for an organization or set of organizations.

The central oversight authority-

1) coordinates protection policy and practices (procedures) documentation,

2) may function as a holder of data provided by service agents, and

3) serves as the source for common and system level information required by service agents (e.g., keying material and registration information, directory data, system policy specifications, and system-wide key compromise and certificate revocation information).

**2.Key Processing Facility(ies):**

Key processing services typically include one or more of the following:

• Acquisition or generation of public key certificates (where applicable),

• Initial generation and distribution of keying material,

• Maintenance of a database that maps user entities to an organization’s certificate/key structure,

• Maintenance and distribution of compromise key lists (CKLs) and/or certificate revocation lists (CRLs), and

• Generation of audit requests and the processing of audit responses as necessary for the prevention of undetected compromises.

**3.Service Agents:**

* Service agents support organizations’ KMIs as single points of access for other KMI nodes. All transactions initiated by client nodes are either processed by a service agent or forwarded to other nodes for processing.
* Service agents direct service requests from client nodes to key processing facilities, and when services are required from multiple processing facilities, coordinate services among the processing facilities to which they are connected.
* Service agents are employed by users to order keying material and services, retrieve keying material and services, and manage cryptographic material and public key certificates.
* A service agent may provide cryptographic material and/or certificates by utilizing specific key processing facilities for key and/or certificate generation.
* A service agent that supports a major organizational unit or geographic region may either access a central or inter-organizational key processing facility or employ local, dedicated processing facilities as required to support survivability, performance, or availability, requirements (e.g., a commercial external Certificate Authority).

**4.Client Nodes:**

* Client nodes are interfaces for managers, devices, and applications to access KMI functions, including the requesting of certificates and other keying material.
* They may include cryptographic modules, software, and procedures necessary to provide user access to the KMI. Client nodes interact with service agents to obtain cryptographic key services.
* Client nodes provide interfaces to end user entities (e.g., encryption devices) for the distribution of keying material, for the generation of requests for keying material, for the receipt and forwarding (as appropriate) of compromised key lists (CKLs) and/or certificate revocation lists (CRLs), for the receipt of audit requests, and for the delivery of audit responses.
* Client nodes typically initiate requests for keying material in order to synchronize new or existing user entities with the current key structure, and receive encrypted keying material for distribution to end-user cryptographic devices (in which the content - the unencrypted keying material – is not usually accessible to human users or user-node interface processes).

**Q.26. Write a short note on key management policy. (Unit-2)**

**Ans:**

* A key management policy is a set of rules that are established to describe the goals, responsibilities, and overall requirements for the management of cryptographic keying material used to protect private or critical facilities, processes, or information.
* Key Management Policies (KMP) are implemented through a combination of security mechanisms and procedures. An organization uses security mechanisms (e.g., safes, alarms, random number generators, encryption algorithms, signature and authentication algorithms) as tools to implement a policy.
* However, key management mechanisms will produce the desired results only if they are properly configured and maintained.
* The KMP is a high-level document that describes authorization and protection objectives and constraints that apply to the generation, distribution, accounting, storage, use, and destruction of cryptographic keying material.

**1.Policy Content:**

* The Key Management Policy (KMP) is a high-level statement of organizational key management policies that includes authorization and protection objectives, and constraints that apply to the generation, distribution, accounting, storage, use, and destruction of cryptographic keying material.
* The policy document or documents that comprise the KMP will include high-level key management structure and responsibilities, governing standards and guidelines, organizational dependencies and other relationships, and security objectives.
* The scope of a KMP may be limited to the operation of a single PKI Certificate Authority (CA) and its supporting components, or to a symmetric point-to-point or single key center environment.
* Alternatively, the scope of a KMP may be the operations of a hierarchical PKI, bridged PKI, or multiple center symmetric key environment.
* The KMP is used for a number of different purposes. The KMP is used to guide the development of KMPSs for each PKI CA or symmetric key management group that operates under its provisions.
* CAs from other organizations’ PKIs may review the KMP before cross-certification, and managers of symmetric key KMIs may review the KMP before joining new or existing multiple center groups.
* Auditors and accreditors will use the KMP as the basis for their reviews of PKI CA and/or symmetric key KMI operations. Application owners that are considering a PKI certificate source **should** review a KMP/CP to determine whether its certificates are appropriate for their applications.

**2**. **Policy Enforcement:**

* In order to be effective, key management policies **shall** be enforced, and policy implementation **should** be evaluated on a regular basis.
* Evaluation requirements will vary with the size and complexity of an organization’s protected communications infrastructure.
* Each organization will need to determine its requirements based on the sensitivity of information being exchanged, the communications volume associated with sensitive or critical information and processes, personnel resources, the size and complexity of the organization or organizations supported, the variety and numbers of cryptographic devices and applications, the types of cryptographic devices and applications, and the scale and complexity of protected communications facilities.

**Q.27. What are the security objectives of key management policy? (Unit-2)**

**Ans:**

A KMP **should** state the security objectives that are applicable to and expected to be supported by the KMI. The security objectives **should** include the identification of:

(a) The nature of the information to be protected (e.g., financial transactions, confidential information, critical process data);

(b) The classes of threats against which protection is required (e.g., the unauthorized modification of data, replay of communications, fraudulent repudiation of transactions, disclosure of information to unauthorized parties);

(c) The Federal Information Processing Standard 199 (FIPS 199) impact level which is determined by the consequences of a compromise of the protected information and/or processes (including sensitivity and perishability of the information);

(d) The cryptographic protection mechanisms to be employed (e.g., message authentication, digital signature, encryption);

(e) Protection requirements for cryptographic processes and keying material (e.g., tamper-resistant processes, confidentiality of keying material); and

(f) Applicable statutes, and executive directives and guidance to which the KMI and its supporting documentation **shall** conform.

**Q.28. Explain the sample KMP format. (Unit-2)**

**Ans:**

**(a) Introduction**- The *Introduction* identifies and introduces the provisions of the policy document and indicates the security objectives and the types of entities and applications for which the KMP is targeted.

This section has the following subsections: 1) Overview, 2) Identification, 3) Community and Applicability, and 4) Contact Details.

**Overview** - This subsection introduces the KMP.

**Objectives** – This subsection states the security objectives applicable to and expected to be supported by the KMI.

**Community and Applicability** - This subsection identifies the types of entities that distribute keys or certificates.

**Contact Details** - This subsection includes the organization, telephone number, and mailing and/or network address of the Keying Material Manager. This is the authority responsible for the registration, maintenance, and interpretation of the KMP.

**(b) General Provisions**–

The *General Provisions* section of the KMP identifies any applicable policies regarding a range of legal and general practices topics. This section may contain subsections covering

1) obligations, 2) liability, 3) financial responsibility, 4) interpretation and enforcement, 5) fees, 6) publication and repositories, 7) compliance audit, 8) confidentiality, and 9) intellectual property rights.

Each subcomponent may need to separately state the provisions applying to each KMI entity type (e.g., central oversight authority, key processing facility, service agent, client node, PKI CA, PKI repository, PKI RA, PKI subscriber, and/or PKI relying party8).

**Obligations** - This subsection contains, for each entity type, any applicable policies regarding the entity's obligations to other entities. Such provisions may include: 1) Keying Material Manager and/or Central Oversight Authority obligations, 2) Key Processing Facility obligations, 3) Service Agent obligations, 4) CA and/or RA obligations (PKI), 4) User obligations (including Client Nodes and PKI subscribers and relying parties), and 5) Keying Material Repository obligations.

**Liability** - This subsection contains, for each entity type, any applicable policies regarding the apportionment of liability (e.g., warranties and limitations on warranties, kinds of damages covered and disclaimers, loss limitations per certificate or per transaction, and other exclusions like acts of God).

**Financial Responsibility** - This subsection contains, for key and/or certificate providers (e.g., key processing facilities, key distribution or translation centers, PKI CAs, key or certificate repositories, PKI RAs), any applicable policies regarding financial responsibilities, such as 1) the indemnification of KMI provider entity relying parties, 2) fiduciary relationships (or lack thereof) among the various entities; and 3) administrative processes (e.g., accounting, audit).

**Interpretation and Enforcement** - This subsection contains any applicable policies regarding the interpretation and enforcement of the KMP or KMPS, addressing such topics as 1) governing law; 2) the severability of provisions, survival, merger, and notice; and 3) dispute resolution procedures.

**Fees** - This subsection contains any applicable policies regarding interagency reimbursement or fees charged by key variable and/or certificate providers (e.g., reimbursement for key center management, certificate issuance or renewal fees, a certificate access fee, revocation or status information access fee, reimbursement for information desk services, fees for other services such as policy information, refund policy).

**Publication and Repositories** - This subsection contains any applicable policies regarding 1) a key and/or certificate source’s obligations to publish information regarding its practices, its products (e.g., keys, certificates), and the current status of such products; 2) the frequency of publication; 3) access control on published information (e.g., policies, practice statements, key variables, certificates, key variable and/or certificate status, CRLs, CKLs); and 4) requirements pertaining to the use of repositories operated by private sector CAs or by other independent parties.

**Compliance Audit** - This subsection addresses any high-level policies regarding 1) the frequency of compliance audit for KMI entities, 2) the identity/qualifications of the auditor, 3) the auditor's relationship to the entity being audited, 4) topics covered under the compliance audit9, 5) actions taken as a result of a deficiency found during compliance audit, 6) the dissemination of compliance audit results.

**Confidentiality Policy** - This subsection states policies regarding 1) the types of information that shallbe kept confidential by KMI entities, 2) the types of information that are not considered confidential, 3) the dissemination of reasons for revocation and suspension of certificates, 4) the release of information to law enforcement officials, 5) information that can be revealed as part of civil discovery, 6) the disclosure of keys or certificates by KMI entities at subscriber/user request; and 7) any other circumstances under which confidential information may be disclosed.

**Intellectual Property Rights** - This subsection addresses policies concerning the ownership rights of certificates, practice/policy specifications, names, and keys.

**(c) Identification and Authentication–**

The *Identification and Authentication* section describes circumstances and identifies any applicable regulatory authority and guidelines regarding the authentication of a certificate applicant or key variable requestor prior to the issuing of key(s) or certificate(s) by a keying material source. This section also includes policies regarding the authentication of parties requesting re-key or revocation. Where applicable, this section also addresses PKI naming practices, including name ownership recognition and name dispute resolution. This section of the KMP has the following subsections:

• Initial Registration,

• Routine Re-key,

• Re-key After Revocation, and

• Revocation Request.

**(d) Operational Requirements –**

The *Operational Requirements* section specifies policies regarding the imposition of requirements on KMI entities with respect to various operational activities. This section may address the following topics:

• Request for shared key variable relationship/Certificate application,

• Initial issuance of key encrypting keys and/or Certificate issuance,

• Acceptance of key variables and Certificates,

• Key and/or Certificate suspension and revocation,

• Security audit requirements,

• Records archiving,

• Key changeover (including re-keying, updating, re-derivation),

• Compromise and disaster recovery, and

• Key Center and/or CA Termination.

Within each topic, separate consideration may need to be given to each KMI entity class.

**(e) Minimum Baseline Security Controls –**

This section states policies regarding management, operational, and technical security controls (i.e., physical, procedural, and personnel controls) used by KMI components to securely perform 1) key generation, 2) subject authentication, 3) key establishment/transfer and/or certificate issuance, 4) key and/or certificate revocation, 5) audit, and 6) archiving.

Based on the FIPS 199 impact level, the appropriate minimum baseline of security controls contained in NIST Special Publication 800-53, *Recommended Security Controls for Federal Information Systems*, **shall** be implemented and described in this section of the KMP.

**(f) Cryptographic Key, Message Interchange, and/or Certificate Formats –**

This section is used to state policies specifying conformance to specific standards and/or guidelines regarding 1) key management architectures and/or protocols, 2) key management message formats, 3) certificate formats and/or 4) CRL/CKL formats.

**(g) Specification and Administration –**

The “Specification Administration” section of the policy document specifies:

• What organization(s) has/have change control responsibility for the KMP,

• Publication and notification procedures for new versions, and

• KMPS approval procedures.

**Q.29. Write a short note on Kerberos.**

**Ans:**

* **Kerberos:**
* **Kerberos** is a [computer network](https://en.wikipedia.org/wiki/Computer_network) [authentication](https://en.wikipedia.org/wiki/Authentication) [protocol](https://en.wikipedia.org/wiki/Cryptographic_protocol) which works on the basis of 'tickets' to allow [nodes](https://en.wikipedia.org/wiki/Node_(networking)) communicating over a non-secure network to prove their identity to one another in a secure manner.
* Its designers aimed it primarily at a [client–server](https://en.wikipedia.org/wiki/Client%E2%80%93server) model and it provides [mutual authentication](https://en.wikipedia.org/wiki/Mutual_authentication)—both the user and the server verify each other's identity. Kerberos protocol messages are protected against [eavesdropping](https://en.wikipedia.org/wiki/Computer_insecurity#Eavesdropping) and [replay attacks](https://en.wikipedia.org/wiki/Replay_attack).
* **Kerberos Authentication:**

Kerberos is a network authentication protocol. It is designed to provide strong authentication for client/server applications by using secret-key cryptography. It has the following characteristics:

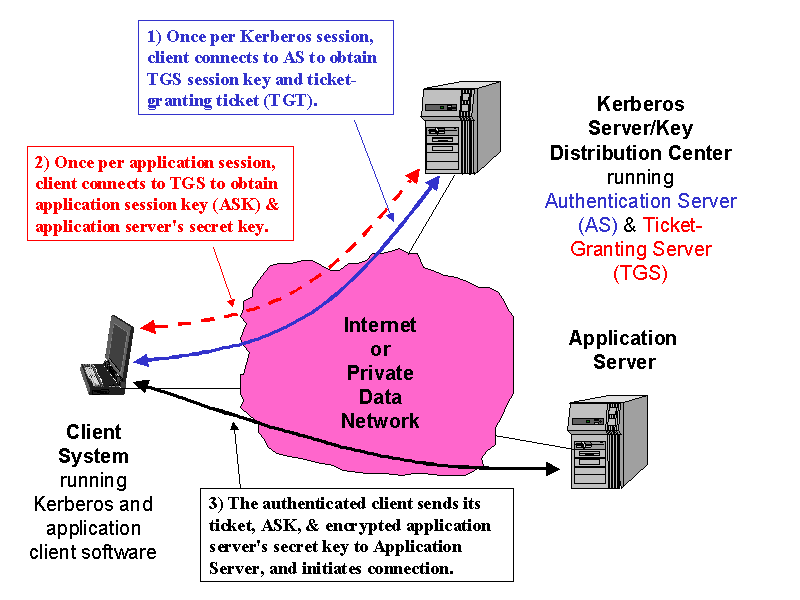
• It is secure: it never sends a password unless it is encrypted.

• Only a single login is required per session. Credentials defined at login are then passed between resources without the need for additional logins.

• The concept depends on a trusted third party – a Key Distribution Center (KDC). The KDC is aware of all systems in the network and is trusted by all of them.

• It performs mutual authentication, where a client proves its identity to a server and a server proves its identity to the client.

* **Kerberos' architecture:**



1.Kerberos server

2.Client processes

3.Application servers

**1.Kerberos' server:**

● executes in a dedicated, physically secure computer

● encompasses:

○ Authentication database:

-> stores secret keys of all system's subjects

○ Authentication server (*AS*):

-> certifies the users' identities

○ Ticket-granting server TGS:

-> supplies clients with tickets to be used with servers

● it is a KDC, *Key Distribution Center*

**2.Client processes:**

● execute on “normal” computers

● the user owning client processes is authenticated only once, on log on

● access to services (from application servers) is only possible by means of ticket supplied from Ticket-granting server

**3.Application servers:**

● execute on networked computers

● each shares a specific key with Ticket-granting server

● only serve clients with tickets proved genuine

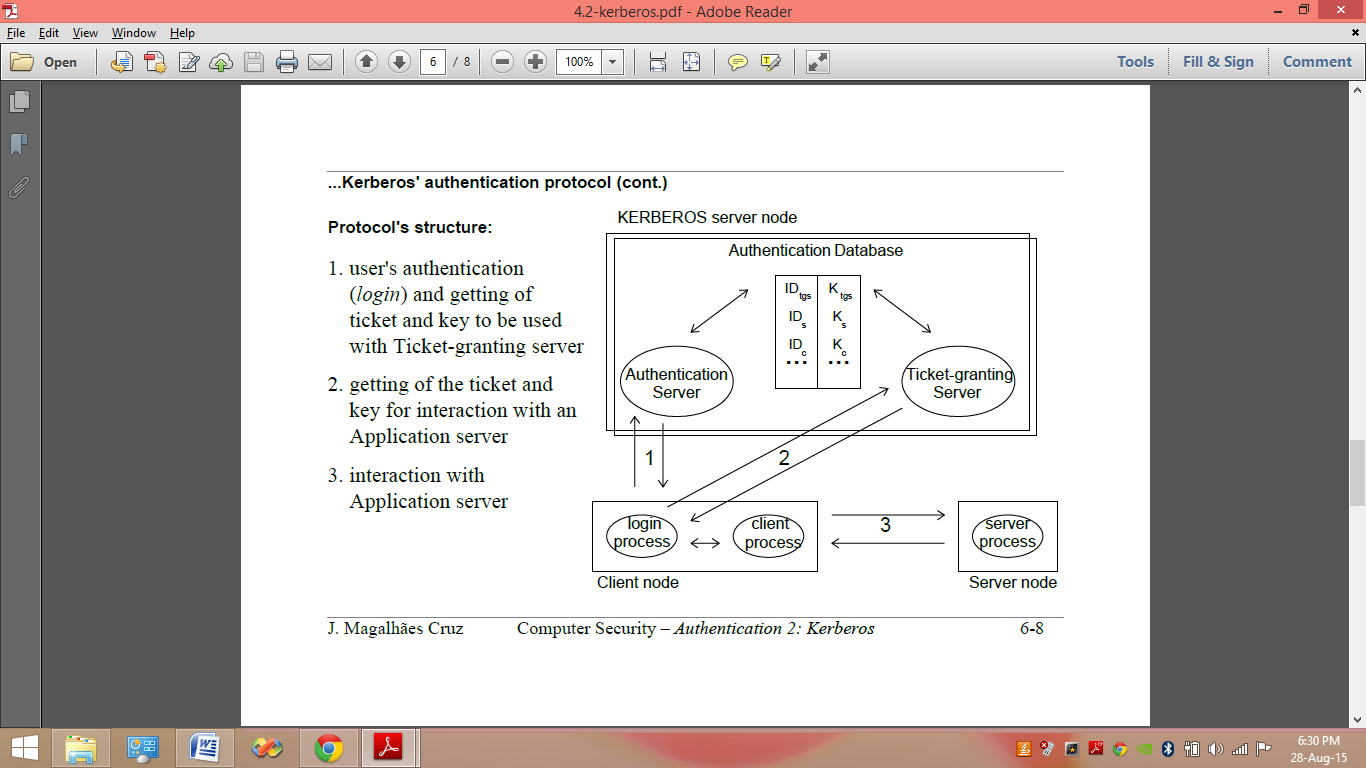
* **Kerberos' authentication protocol :**

**Protocol's structure:**

1. user's authentication (*login*) and getting of ticket and key to be used with Ticket- granting server

2. getting of the ticket and key for interaction with an Application server

3. interaction with Application server

****

**Q.30. List & explain the KMI components in detail.**

**Ans:**

**Refer Q.25.**

**Q.31. Write a short note on Key Management Policy.**

**Ans:**

**Refer Q.26**

**Q.32. Explain any six server security principles.**

**Ans:**

**Q.33. How the server security is planned?**

**Ans:**

**Q.34. How the server security is maintained?**

**Ans:**