**1.A Apply access control concepts, methodologies, and techniques**

**1.A.1 Application control concepts and principles (e.g. discretionary/mandatory, segregation/separation of duties, rule of least privilege)**

**DAC**

Known as **Identity-based Access control** - The type of discretionary access control (DAC) that is based on an individual's identity

With Discretionary Access Control, the subject has authority, within certain limitations, to specify what objects can be accessible.

For example, access control lists can be used. This type of access control is used in local, dynamic situations where the subjects must have the discretion to specify what resources certain users are permitted to access.

When a user, within certain limitations, has the right to alter the access control to certain objects, this is termed as user-directed discretionary access control. In some instances, a hybrid approach is used, which combines the features of user-based and identity-based discretionary access control.

**The read privilege** is the most problematic privilege regarding information flows. The privilege essentially allows the subject to create a copy of the object in memory.

**Non-DAC -> more DAC than MAC**

Non-DAC access control is when a **central authority determines what subjects** can have access to what objects based on the organizational security policy. Centralized access control is not an existing security model.

• **RBAC -** include Role Based Access Control (RBAC) and Rule Based Access Control (RBAC or RuBAC). RABC being a subset of NDAC, it was easy to eliminate RBAC as it was covered under NDAC already.

• **TBAC – Task based, Time – Sequence - Dependencies**

**• ORCON – Originator controlled – owner controls the lifecycle**

**• DRM – digital rights management, intellectual content**

**• UCON – Usage controlled – frequency of access.**

**• Rule-based -** is when you have ONE set of rules applied uniformly to all users. A good example would be a firewall at the edge of your network. A single rule based is applied against any packets received from the internet.

**MAC -** is a very rigid type of access control. The subject must dominate the object and the subject must have a Need To Know to access the information. Objects have labels that indicate the sensitivity (classification) and there is also categories to enforce the Need To Know (NTK).

- Classifications and labels

- Confidential, Secret, Top Secret

- A Sensitivity label must contain at least one classification and one category set.

- Category set = compartment set = The sensitivity label must contain at least one Classification and at least one Category Set/Compartment set, but it is common in some environments for a single item to belong to multiple category sets. The list of all the categories to which an item belongs is called a compartment set or category set.

MAC provides high security by regulating access based on the clearance of individual users and sensitivity labels for each object. Clearance levels and sensitivity levels cannot be modified by individual users -- for example, user Joe (SECRET clearance) cannot reclassify the "Presidential Doughnut Recipe" from "SECRET" to "CONFIDENTIAL" so that his friend Jane (CONFIDENTIAL clearance) can read it. The administrator is ultimately responsible for configuring this protection in accordance with security policy and directives from the Data Owner.

**A Sensitivity label must contain at least one classification and one category set.**

In a Mandatory Access Control (MAC) environment active entities that take actions are called Subjects and the passive entities that are acted upon are known as Objects. A **Subject could be a users, a programs, and processes** where **Objects would be files, directories, print queue, devices, windows, and sockets.**

All of those listed in the answer are defined as Objects.

Note: Some devices could be considered active entities such as smart card readers and some devices such as electrical switches, disc drives, relays, and mechanical components connected to a computer system may also be included in the category of objects [NCSC88].

This makes this answer the best based on the other answers.

**• Lattice-based access control** - In this type of control, a lattice model is applied. To apply this concept to access control, the pair of elements is the subject and object, and the subject has to have an upper bound equal or higher than the object being accessed.

**NOTE FROM CLEMENT:**

Lot of people tend to confuse MAC and Rule Based Access Control.

Mandatory Access Control must make use of LABELS. If there is only rules and no label, it cannot be Mandatory Access Control. This is why they call it Non Discretionary Access control (NDAC).

There are even books out there that are WRONG on this subject. Books are sometimes opiniated and not strictly based on facts.

In MAC subjects must have clearance to access sensitive objects. Objects have labels that contain the classification to indicate the sensitivity of the object and the label also has categories to enforce the need to know.

Today the best example of rule based access control would be a firewall. All rules are imposed globally to any user attempting to connect through the device. This is NOT the case with MAC.

**Bell Lapadula**

The Bell-LaPadula model is a formal model dealing with confidentiality.

The Bell-LAPadula model is also called a multilevel security system because users **with different clearances use** the system and the system processes data with different classifications. Developed by the US Military in the 1970s.

The Bell–LaPadula Model (abbreviated BLP) is a state machine model used for enforcing access control in government and military applications. It was developed by David Elliott Bell and Leonard J. LaPadula, subsequent to strong guidance from Roger R. Schell to formalize the U.S. Department of Defense (DoD) multilevel security (MLS) policy. The model is a formal state transition model of computer security policy that describes a set of access control rules which use security labels on objects and clearances for subjects. Security labels range from the most sensitive (e.g."Top Secret"), down to the least sensitive (e.g., "Unclassified" or "Public").

The Bell–LaPadula model focuses on data confidentiality and controlled access to classified information, in contrast to the Biba Integrity Model which describes rules for the protection of data integrity. In this formal model, the entities in an information system are divided into subjects and objects.

The notion of a "secure state" is defined, and it is proven that each state transition preserves security by moving from secure state to secure state, thereby inductively proving that the system satisfies the security objectives of the model. The Bell–LaPadula model is built on the concept of a state machine with a set of allowable states in a computer network system. The transition from one state to another state is defined by transition functions.

A system state is defined to be "secure" if the only permitted access modes of subjects to objects are in accordance with a security policy. To determine whether a specific access mode is allowed, the clearance of a subject is compared to the classification of the object (more precisely, to the combination of classification and set of compartments, making up the security level) to determine if the subject is authorized for the specific access mode.

The clearance/classification scheme is expressed in terms of a lattice. The model defines two mandatory access control (MAC) rules and one discretionary access control (DAC) rule with three security properties:

**The Star (\*) property (confinement property) rule of the Bell-LaPadula model says that subjects NO WRITE DOWN**, this would compromise the confidentiality of the information if someone at the secret layer would write the object down to a confidential container for example.

**The Simple Security (ss) Property rule states that the subject NO READ UP** which means that a subject at the secret layer would not be able to access objects at Top Secret for example.

The Bell-LaPadula model, mostly concerned with confidentiality, was proposed for enforcing access control in government and military applications. It supports mandatory access control by determining the access rights from the security levels associated with subjects and objects. It also supports discretionary access control by checking access rights from an access matrix. The Biba model, introduced in 1977, the Sutherland model, published in 1986, and the Brewer-Nash model, published in 1989, are concerned with integrity.

Tranquility principle - The tranquility principle of the Bell-LaPadula model states that the classification of a subject or object does not change while it is being referenced. There are two forms to the tranquility principle: the "principle of strong tranquility" states that security levels do not change during the normal operation of the system and the "principle of weak tranquility" states that security levels do not change in a way that violates the rules of a given security policy. Another interpretation of the tranquility principles is that they both apply only to the period of time during which an operation involving an object or subject is occurring. That is, the strong tranquility principle means that an object's security level/label will not change during an operation (such as read or write); the weak tranquility principle means that an object's security level/label may change in a way that does not violate the security policy during an operation.

**Clark-Wilson** model addresses integrity. It incorporates mechanisms to enforce internal and external consistency, a separation of duty, and a mandatory integrity policy.

In the Clark-Wilson model, the subject no longer has direct access to objects but instead must access them through programs (well -formed transactions).

The Clark-Wilson model uses separation of duties, which divides an operation into different parts and requires different users to perform each part. This prevents authorized users from making unauthorized modifications to data, thereby protecting its integrity.

The Clark-Wilson integrity model provides a foundation for specifying and analyzing an integrity policy for a computing system.

The model is primarily concrned with formalizing the notion of information integrity. Information integrity is maintained by preventing corruption of data items in a system due to either error or malicious intent. An integrity policy describes how the data items in the system should be kept valid from one state of the system to the next and specifies the capabilities of various principals in the system. The model defines enforcement rules and certification rules.

The model’s enforcement and certification rules define data items and processes that provide the basis for an integrity policy. The core of the model is based on the notion of a transaction.

A well-formed transaction is a series of operations that transition a system from one consistent state to another consistent state.

In this model the integrity policy addresses the integrity of the transactions.

The principle of separation of duty requires that the certifier of a transaction and the implementer be different entities.

The model contains a number of basic constructs that represent both data items and processes that operate on those data items. The key data type in the Clark-Wilson model is a Constrained Data Item (CDI). An Integrity Verification Procedure (IVP) ensures that all CDIs in the system are valid at a certain state. Transactions that enforce the integrity policy are represented by Transformation Procedures (TPs). A TP takes as input a CDI or Unconstrained Data Item (UDI) and produces a CDI. A TP must transition the system from one valid state to another valid state. UDIs represent system input (such as that provided by a user or adversary). A TP must guarantee (via certification) that it transforms all possible values of a UDI to a “safe” CDI.

**Biba -** The **simple integrity axiom** of the Biba access control model states that a subject at one level of integrity is not permitted to observe an object of a lower integrity **(no read down).**

BIBA addresses only the first rule of integrity.

**Least Privilege**

Does not require: Ensuring that the user alone does not have sufficient rights to subvert an important process.

- Least Functionality

In the real world, organizations are understaffed, people wear too many hats, separation of duties is seldom complete, development staff often supports production systems, IT staff often maintains production data and access is often granted on the basis of **least effort** instead of least privilege and need to know.

**Separation of Duties**

Preventive control

**Architecture**

**- Trusted Computing Base**

**o Reference** Monitor – get requests and check if it was allowed.

**o Kernel –** a collection of components of TCB.

**1.A.2 Account life cycle management (e.g. registration, enrolment, access control administration)**

**1.A.3 Identification, authentication, authorization, and accounting methods**

**Authentication** - is verification that the user's claimed identity is valid and is usually implemented through a user password at log-on time.

Accountability would not include policies & procedures because while important on an effective security program they cannot be used in determing accountability.

**Accountability**  = access rules + audit trails + unique identifiers.

**Identification** - act of a user professing an identity to a system, usually in the form of a log-on ID

A **continuous authentication** provides protection against impostors who can see, alter, and insert information passed between the claimant and verifier even after the claimant/verifier authentication is complete. This is the best **protection against hijacking**. Static authentication is the type of authentication provided by traditional password schemes and the strength of the authentication is highly dependent on the difficulty of guessing passwords. The robust authentication mechanism relies on dynamic authentication data that changes with each authenticated session between a claimant and a verifier, and it does not protect against hijacking. Strong authentication refers to a two-factor authentication (like something a user knows and something a user is).

The important thing to remember is that **Identity Management Solutions** provides **consistency, efficiency, usability, reliability and scalabilty.**

FACEBOOK, TWEETER -> Any identity management system used in an environment where there are tens of thousands of users must be able to **scale to support the volumes of data and peak transaction rates.**

**BIOMETRIC:**

When used in physical control biometric Identification is performed by doing a one to many match. When you submit your biometric template a search is done through a database of templates until the matching one is found. At that point your identity is revealed and if you are a valid employee access is granted. When used in logical controls the biometric template is used to either confirm or deny someone identity. For example if I access a system and I pretend to be user Nathalie then I would provide my biometric template to confirm that I really am who I pretend to be. Biometric is one of the three authentication factor (somethin you are) that can be use. The other two are something you know and something you have.

**RETINA -** **least accepted, most expensive!**

**IRIS pattern** is considered lifelong. Unique features of the iris are: freckles, rings, rifts, pits, striations, fibers, filaments, furrows, vasculature and coronas. Voice, signature and retina patterns are more likely to change over time, thus are not as suitable for authentication over a long period of time without needing re-enrollment.

Because the optical unit utilizes a camera and infrared light to create the images, sun light can impact the aperture so it must not be positioned in direct light of any type. Because the subject does not need to have direct contact with the optical reader, direct light can impact the reader.

An Iris recognition is a form of biometrics that is based on the uniqueness of a subject's iris. A camera like device records the patterns of the iris creating what is known as Iriscode.

It is the unique patterns of the iris that allow it to be one of the most accurate forms of biometric identification of an individual. Unlike other types of biometics, the iris rarely changes over time. Fingerprints can change over time due to scaring and manual labor, voice patterns can change due to a variety of causes, hand geometry can also change as well. But barring surgery or an accident it is not usual for an iris to change. The subject has a high-resoulution image taken of their iris and this is then converted to Iriscode. The current standard for the Iriscode was developed by John Daugman. When the subject attempts to be authenticated an infrared light is used to capture the iris image and this image is then compared to the Iriscode. If there is a match the subject's identity is confirmed. The subject does not need to have direct contact with the optical reader so it is a less invasive means of authentication then retinal scanning would be.

When a biometric system is used for access control, the most important error is the false accept or false acceptance rate, or Type II error, where the system would accept an impostor.

A Type I error is known as the false reject or false rejection rate and is not as important in the security context as a type II error rate. A type one is when a valid company employee is rejected by the system and he cannot get access even thou it is a valid user.

The Crossover Error Rate (CER) is the point at which the false rejection rate equals the false acceptance rate if your would create a graph of Type I and Type II errors. The lower the CER the better the device would be.

**EER or CER -** Equal error rate or crossover error rate (EER or CER): the rate at which both accept and reject errors are equal. The value of the EER can be easily obtained from the ROC curve. The EER is a quick way to compare the accuracy of devices with different ROC curves. In general, the device with the lowest EER is most accurate.

The Crossover Error Rate (CER) is the point where false rejection rate (type I error) equals the false acceptance rate (type II error). The lower the CER, the better the accuracy of the device. At the time if this writing, response times and accuracy of some devices are:

**System type Response time Accuracy (CER)**

**Fingerprints 5-7 secs. 5%**

**Hand Geometry 3-5 secs. 2%**

**Voice Pattern 10-14 secs. 10%**

**Retina Scan 4-7 secs. 1.5%**

**Iris Scan 2.5-4 secs. 0.5%**

The term EER which means Equal Error Rate is sometimes use instead of the term CER. It has the same meaning.

In the context of Biometric Authentication almost all types of detection permit a system's sensitivity to be increased or decreased during an inspection process. If the system's sensitivity is increased, such as in an airport metal detector, the system becomes increasingly selective and has a higher False Reject Rate (FRR).

Conversely, if the sensitivity is decreased, the False Acceptance Rate (FAR) will increase.

Thus, to have a valid measure of the system performance, the CrossOver Error Rate (CER) is used.

The following are used as performance metrics for biometric systems:

False accept rate or false match rate (FAR or FMR): the probability that the system incorrectly matches the input pattern to a non-matching template in the database. It measures the percent of invalid inputs which are incorrectly accepted. In case of similarity scale, if the person is imposter in real, but the matching score is higher than the threshold, then he is treated as genuine that increase the FAR and hence performance also depends upon the selection of threshold value.

**False reject rate or false non-match rate (FRR or FNMR):** the probability that the system fails to detect a match between the input pattern and a matching template in the database. It measures the percent of valid inputs which are incorrectly rejected.

**Failure to enroll rate (FTE or FER):** the rate at which attempts to create a template from an input is unsuccessful. This is most commonly caused by low quality inputs.

**Failure to capture rate (FTC):** Within automatic systems, the probability that the system fails to detect a biometric input when presented correctly.

**Template capacity:** the maximum number of sets of data which can be stored in the system.

Almost all types of detection permit a system's sensitivity to be increased or decreased during an inspection process. If the system's sensitivity is increased, such as in a biometric authentication system, the system becomes increasingly selective and has a higher False Rejection Rate (FRR).

Conversely, if the sensitivity is decreased, the False Acceptance Rate (FRR) will increase. Thus, to have a valid measure of the system performance, the Cross Over Error (CER) rate is used. The Crossover Error Rate (CER) is the point at which the false rejection rates and the false acceptance rates are equal. The lower the value of the CER, the more accurate the system.

**Throughput -** The throughput rate is the rate at which individuals, once enrolled, can be processed and identified or authenticated by a biometric system.

Acceptable throughput rates are in the range of 10 subjects per minute.

Things that may impact the throughput rate for some types of biometric systems may include:

A concern with retina scanning systems may be the exchange of body fluids on the eyepiece.

Another concern would be the retinal pattern that could reveal changes in a person's health, such as diabetes or high blood pressure.

**Masquarading attacks -** The only way to be truly positive in authenticating identity for access is to base the authentication on the physical attributes of the persons themselves (i.e., biometric identification). Physical attributes cannot be shared, borrowed, or duplicated. They ensure that you do identify the person, however they are not perfect and they would have to be supplemented by another factor.

Some people are getting thrown off by the term Masquarade. In general, a masquerade is a disguise. In terms of communications security issues, a masquerade is a type of attack where the attacker pretends to be an authorized user of a system in order to gain access to it or to gain greater privileges than they are authorized for. A masquerade may be attempted through the use of stolen logon IDs and passwords, through finding security gaps in programs, or through bypassing the authentication mechanism. Spoofing is another term used to describe this type of attack as well.

A UserId only provides for identification.

A password is a weak authentication mechanism since passwords can be disclosed, shared, written down, and more.

A smart card can be stolen and its corresponding PIN code can be guessed by an intruder. A smartcard can be borrowed by a friend of yours and you would have no clue as to who is really logging in using that smart card.

Any form of two-factor authentication not involving biometrics cannot be as reliable as a biometric system to identify the person.

See an extract below from the HISM book volume 1 (Available on www.cccure.org in HTML format)

Biometric identifying verification systems control people. If the person with the correct hand, eye, face, signature, or voice is not present, the identification and verification cannot take place and the desired action (i.e., portal passage, data, or resource access) does not occur.

As has been demonstrated many times, adversaries and criminals obtain and successfully use access cards, even those that require the addition of a PIN. This is because these systems control only pieces of plastic (and sometimes information), rather than people. Real asset and resource protection can only be accomplished by people, not cards and information, because unauthorized persons can (and do) obtain the cards and information.

Further, life-cycle costs are significantly reduced because no card or PIN administration system or personnel are required. The authorized person does not lose physical characteristics (i.e., hands, face, eyes, signature, or voice), but cards and PINs are continuously lost, stolen, or forgotten. This is why card access systems require systems and people to administer, control, record, and issue (new) cards and PINs. Moreover, the cards are an expensive and recurring cost.

Authentication is based on three factor types: type 1 is something you know, type 2 is something you have and type 3 is something you are. Biometrics are based on the Type 3 authentication mechanism.

**Most effective from Best to worst:**

**Retina**

**Fingerprint**

**Hand Print**

**Voice pattern**

**Keystroke**

**Signature**

**Order of acceptance from most accepted to least accepted:**

**Keystroke**

**Signature**

**Voice**

**Handprint**

**Fingerprint**

**Retina**

In **biometrics**, identification is a **"one-to-many"** search of an individual's characteristics from a database of stored images.

**1.B Determine identity and access management architecture**

Authentication, Authorization and Accounting - AAA

**1.B.1 Centralized**

Here one entity (dept or an individual) is responsible for overseeing access to all corporate resources.

This type of administration provides a consistent and uniform method of controlling users access rights.

Example: RADIUS, TACACS and Diameter

**1.B.2 Decentralized**

A decentralized access control administration method gives control of access to the people closer to the resources

In this approach, it is often the functional manager who assigns access control rights to employees.

Changes can happen faster through this type of administration because not just one entity is making changes for the whole organization.

There is a possibility for conflicts to arise that may not benefit the organization as because different managers and departments can practice security and access control in different ways.

There is a possibility of certain controls to overlap, in which case actions may not be properly proscribed or restricted.

This type of administration does not provide methods for consistent control, as a centralized method would.

MS Active Directory, OpenLDAP, etc.…

**1.B.3 Federated Identity**

in information technology is the means of linking a person's electronic identity and attributes, stored across multiple distinct identity management systems.[1]

Related to federated identity is single sign-on (SSO), in which a user's single authentication ticket, or token, is trusted across multiple IT systems or even organizations. SSO is a subset of federated identity management, as it relates only to authentication and is understood on the level of technical interoperability.

Technologies used for federated logon include **SAML (Security Assertion Markup Language), OAuth and OpenID** as well as proprietary standards.

**1.B.4 Access Control Protocols and Technologies:**

**TACACS**

The original TACACS, developed in the early ARPANet days, had very limited functionality and used the UDP transport.

In the early 1990s, the protocol was extended to include additional functionality and the transport changed to TCP.

It is a remote authentication protocol that is used to communicate with an authentication server commonly used in UNIX networks. TACACS allows a remote access server to communicate with an authentication server in order to determine if the user has access to the network. TACACS is defined in RFC 1492, and uses (either TCP or UDP) port 49 by default.

TACACS allows a client to accept a username and password and send a query to a TACACS authentication server, sometimes called a TACACS daemon or simply TACACSD. TACACSD uses TCP and usually runs on port 49. It would determine whether to accept or deny the authentication request and send a response back. The TIP (routing node accepting dial-up line connections, which the user would normally want to log in into) would then allow access or not, based upon the response. In this way, the process of making the decision is "opened up" and the algorithms and data used to make the decision are under the complete control of whomever is running the TACACS daemon.

A later version of TACACS introduced by Cisco in 1990 was called Extended TACACS (XTACACS). The XTACACS protocol was developed by and is proprietary to Cisco Systems.

TACACS+ and RADIUS have generally replaced TACACS and XTACACS in more recently built or updated networks. TACACS+ is an entirely new protocol and is not compatible with TACACS or XTACACS. TACACS+ uses the Transmission Control Protocol (TCP) and RADIUS uses the User Datagram Protocol (UDP). Since TCP is connection oriented protocol, TACACS+ does not have to implement transmission control. RADIUS, however, does have to detect and correct transmission errors like packet lost, timeout etc. since it rides on UDP which is connectionless.

RADIUS encrypts only the users' password as it travels from the RADIUS client to RADIUS server. All other information such as the username, authorization, accounting are transmitted in clear text. Therefore it is vulnerable to different types of attacks. TACACS+ encrypts all the information mentioned above and therefore does not have the vulnerabilities present in the RADIUS protocol.

**TACACS+ -** is an access control network protocol for routers, network access servers and other networked computing devices.

Unlike RADIUS and the predecessors of TACACS+ (TACACS and XTACACS), TACACS+ provides separate authentication, authorization and accounting services. Like RADIUS, TACACS and XTACACS, TACACS+ is an open, publicly documented protocol. TACACS+ uses the TCP protocol and encrypts the entire packet (except the header).

TACACS+ ultimately derives from (but is not backwards-compatible with) TACACS, developed in 1984 for MILNET by BBN, a contractor for the U.S. Department of Defense. Originally designed as means to automate logins, by which a person who was already authenticated on one host in the network could connect to another host on the same network without needing to authenticate again, TACACS is an open (quasi-)standard, described by BBN's Brian Anderson in Internet Engineering Task Force (IETF) RFC 927.

Cisco Systems began supporting TACACS in its networking products in the late 1980s, eventually adding their own extensions to the protocol, which the company then called 'XTACACS' ('eXtended TACACS'). In the simple (non-extended) form, Cisco's implementation was compatible with the original TACACS, while the extended form (XTACACS) was not. In 1993, with Cisco's assistance, Craig Finseth of the University of Minnesota published a description of Cisco's extensions in IETF RFC 1492.

The protocol continued to evolve over the following years, and in 1996 became what Cisco called 'TACACS+', in which the individual tasks of authentication, authorization and accounting were separate processes. Also, while the XTACACS and TACACS use UDP (port 49), TACACS+ uses TCP (but still port 49). Cisco's David Carrel and Lol Grant submitted TACACS+ v1.75 for IETF standards approval in October 1996, followed by a revised version 1.78 in January 1998. The IETF draft expired in September 1998 without becoming an approved standard.

TACACS+ and RADIUS have generally replaced the older protocols.

Whereas RADIUS combines authentication and authorization in a user profile, TACACS+ separates the two operations. Another difference is that TACACS+ uses the Transmission Control Protocol (TCP) while RADIUS uses the User Datagram Protocol (UDP).

The extensions to the TACACS+ protocol provide for more types of authentication requests and more types of response codes than were in the original specification.

TACACS has been through three generations: TACACS, Extended TACACS (XTACACS), and TACACS+.

TACACS combines its authentication and authorization processes,

**XTACACS separates authentication, authorization, and auditing processes and**

TACACS+ is XTACACS with extended two-factor user authentication.

**RADIUS -**

A Network Access Server (NAS) operates as a client of RADIUS. The client is responsible for passing user information to designated RADIUS servers, and then acting on the response which is returned. RADIUS servers are responsible for receiving user connection requests, authenticating the user, and then returning all configuration information necessary for the client to deliver service to the user. RADIUS authentication is based on provisions of simple username/password credentials. These credentials are encrypted by the client using a shared secret between the client and the RADIUS server. OIG 2007, Page 513 RADIUS incorporates an authentication server and can make uses of both dynamic and static passwords.

Since it uses the PAP and CHAP protocols, it also incluses static passwords.

RADIUS is an Internet protocol. RADIUS carries authentication, authorization, and configuration information between a Network Access Server and a shared Authentication Server. RADIUS features and functions are described primarily in the IETF (International Engineering Task Force) document RFC2138.

The term " RADIUS" is an acronym which stands for Remote Authentication Dial In User Service.

The main advantage to using a RADIUS approach to authentication is that it can provide a stronger form of authentication. RADIUS is capable of using a strong, two-factor form of authentication, in which users need to possess both a user ID and a hardware or software token to gain access.

Token-based schemes use dynamic passwords. Every minute or so, the token generates a unique 4-, 6- or 8-digit access number that is synchronized with the security server. To gain entry into the system, the user must generate both this one-time number and provide his or her user ID and password.

Although protocols such as RADIUS cannot protect against theft of an authenticated session via some realtime attacks, such as wiretapping, using unique, unpredictable authentication requests can protect against a wide range of active attacks.

RADIUS supports dynamic passwords and challenge/response passwords.

RADIUS allows the user to have a single user ID and password for all computers in a network.

In response to an access-request from a client, a RADIUS server returns one of three authentication responses: **access-accept, access-reject, or access-challenge**, the latter being a request for additional authentication information such as a one-time password from a token or a callback identifier.

**Diameter uses SCTP or TCP while RADIUS typically uses UDP as the transport layer. As of 2012, RADIUS can also use TCP or TLS as the transport layer.**

* **A RADIUS server can act as a proxy server, forwarding client requests to other authentication domains.**
* **Most of RADIUS clients have a capability to query secondary RADIUS servers for redundancy.**
* **Most RADIUS servers have built-in database connectivity for billing and reporting purposes.**

**Kerberos**

addresses the confidentiality and integrity of information.

It also addresses **primarily authentication** but does not directly address availability.

Kerberos depends on Secret Keys or Symmetric Key cryptography. Kerberos a third party authentication protocol. It was designed and developed in the mid 1980's by MIT. It is considered open source but is copyrighted and owned by MIT. It relies on the user's secret keys. The password is used to encrypt and decrypt the keys.

Kerberos is a trusted, credential-based, third-party authentication protocol that was developed at MIT and that uses symmetric (secret) key cryptography to authenticate clients to other entities on a network for access to services.

Kerberos does not use X.509 certificates, which are used in public key cryptography.

Kerberos DOES NOT use a database to keep a copy of all users' public keys.

Kerberos is a trusted, credential-based, third-party authentication protocol that uses symmetric (secret) key cryptography to provide robust authentication to clients accessing services on a network.

One weakness of Kerberos is its Key Distribution Center (KDC), which represents a single point of failure. The KDC contains a database that holds a copy of all of the symmetric/secret keys for the principals.

The server also checks the authenticator and, if that timestamp is valid, it provides the requested service to the client.

Even if the user principal is present in a ticket and only the application server can extract and possibly manage such information (since the ticket is encrypted with the secret key of the service), this is not enough to guarantee the authenticity of the client.

An impostor could capture (remember the hypothesis of an open and insecure network) the ticket when it is sent by a legitimate client to the application server, and at an opportune time, send it to illegitimately obtain the service.

On the other hand, including the IP addresses of the machine from where it is possible to use it is not very useful: it is known that in an open and insecure network addresses are easily falsified. To solve the problem, one has to exploit the fact that the client and server, at least during a session have the session key in common that only they know (also the KDC knows it since it generated it, but it is trusted by definition!!!).

Thus the following strategy is applied: along with the request containing the ticket, the client adds another packet (the authenticator) where the user principal and time stamp (its at that time) are included and encrypts it with the session key; the server which must offer the service, upon receiving this request, unpacks the first ticket, extracts the session key and, if the user is actually who he/she says, the server is able to unencrypt the authenticator extracting the timestamp.

If the latter differs from the server time by less than 2 minutes (but the tolerance can be configured) then the authentication is successful. This underlines the criticality of synchronization between machines belonging to the same realm.

STEP BY STEP: Principle P1 authenticates to the Key Distribution Center (KDC), principle P1 receives a Ticket Granting Ticket (TGT), and principle P1 requests a service ticket from the Ticket Granting Service (TGS) in order to access the application server P2.

The Key Distribution Center (KDC) holds all users' and services' cryptographic keys. It provides authentication services, as well as key distribution functionality. The Authentication Service is the part of the KDC that authenticates a principal.

A Kerberos ticket is issued by a trusted third party. Is is an encrypted data structure that includes the service encryption key. In that sense it is similar to a public-key certificate. However, the ticket is not the key.

Kerberos is **Vulnerable to Password Guessing.**

**The Replay (Playback) Attack**

A replay attack occurs when an intruder steals the packet and presents it to the service as if the intruder were the user. The user's credentials are there -- everything needed to access a resource. This is mitigated by the features of the "Authenticator," which is illustrated in the picture below.

The Authenticator is created for the AS\_REQ or the TGS\_REQ and sends additional data, such as an encrypted IP list, the client's timestamp and the ticket lifetime. If a packet is replayed, the timestamp is checked. If the timestamp is earlier or the same as a previous authenticator, the packet is rejected because it's a replay. In addition, the time stamp in the Authenticator is compared to the server time. It must be within five minutes (by default in Windows).

Kerberos Authenticator to prevent replay attacks

The Authenticator mitigates the Possibility of a replay attack.

If the time skew is greater than five minutes the packet is rejected. This limits the number of possible replay attacks. While it is technically possible to steal the packet and present it to the server before the valid packet gets there, it is very difficult to do.

It's fairly well known that all computers in a Windows domain must have system times within five minutes of each other. This is due to the Kerberos requirement.

**SESAME -** is an authentication and access control protocol, that also supports communication confidentiality and integrity. It provides public key based authentication along with the Kerberos style authentication, that uses symmetric key cryptography. Sesame supports the Kerberos protocol and adds some security extensions like public key based authentication and an ECMA-style Privilege Attribute Service. The complete Sesame protocol is a two step process. In the first step, the client successfully authenticates itself to the Authentication Server and obtains a ticket that can be presented to the Privilege Attribute Server. In the second step, the initiator obtains proof of his access rights in the form of Privilege Attributes Certificate (PAC). The PAC is a specific form of Access Control Certificate as defined in the ECMA-219 document. This document describes the extensions to Kerberos for public key based authentication as adopted in Sesame.

SESAME uses Attribute Certificate (AC) that allows for granular access control . It supports authentication, confidentiality but also authorization. In environment with well defined roles and capability is an issue , SESAME and PERMIS are role based single sign on technologies.

Secure European System for Applications in a Multi-Vendor Environment (Sesame) is a **single-sign-on technology** developed to extend Kerberos functionality and improve upon its weaknesses. Sesame uses asymmetric and symmetric cryptographic technologies to authenticate subjects to network resources.

**Vulnerable to Password Guessing.** The Basic Mechanism in Sesame for strong authentication is as follow:

The user sends a request for authentication to the Authentication Server as in Kerberos, except that SESAME is making use of public key cryptography for authentication where the client will present his digital certificate and the request will be signed using a digital signature. The signature is communicated to the authentication server through the preauthentication fields. Upon receipt of this request, the authentication server will verifies the certificate, then validate the signature, and if all is fine the AS will issue a ticket granting ticket (TGT) as in Kerberos. This TGT will be use to communicate with the privilage attribute server (PAS) when access to a resource is needed.

Users may authenticate using either a public key pair or a conventional (symmetric) key. If public key cryptography is used, public key data is transported in preauthentication data fields to help establish identity.

Kerberos uses tickets for authenticating subjects to objects and SESAME uses Privileged Attribute Certificates (PAC), which contain the subject’s identity, access capabilities for the object, access time period, and lifetime of the PAC. The PAC is digitally signed so that the object can validate that it came from the trusted authentication server, which is referred to as the privilege attribute server (PAS). The PAS holds a similar role as the KDC within Kerberos. After a user successfully authenticates to the authentication service (AS), he is presented with a token to give to the PAS. The PAS then creates a PAC for the user to present to the resource he is trying to access.

**KryptoKnight -** is a Peer to Peer authentication protocol incorporated into the NetSP product from IBM.

**EAP** - framework that supports multiple, optional authentication mechanisms for PPP, including cleartext passwords, challenge-response, and arbitrary dialog sequences. It is intended for use primarily by a host or router that connects to a PPP network server via switched circuits or dial-up lines. It is an authentication framework frequently used in wireless networks and Point-to-Point connections. It is defined in RFC 3748, which made RFC 2284 obsolete, and was updated by RFC 5247.

EAP is an authentication framework providing for the transport and usage of keying material and parameters generated by EAP methods.[1] There are many methods defined by RFCs and a number of vendor specific methods and new proposals exist. EAP is not a wire protocol; instead it only defines message formats. Each protocol that uses EAP defines a way to encapsulate EAP messages within that protocol's messages.

EAP is in wide use. For example, in IEEE 802.11 (WiFi) the WPA and WPA2 standards have adopted IEEE 802.1X with five EAP types as the official authentication mechanisms.

EAP is an authentication framework, not a specific authentication mechanism.[1] It provides some common functions and negotiation of authentication methods called EAP methods. There are currently about 40 different methods defined. Methods defined in IETF RFCs include EAP-MD5, EAP-POTP, EAP-GTC, EAP-TLS, EAP-IKEv2, EAP-SIM, EAP-AKA. Additionally a number of vendor-specific methods and new proposals exist. Commonly used modern methods capable of operating in wireless networks include EAP-TLS, EAP-SIM, EAP-AKA, LEAP and EAP-TTLS. Requirements for EAP methods used in wireless LAN authentication are described in RFC 4017.

The standard also describes the conditions under which the AAA key management requirements described in RFC 4962 can be satisfied.

RFC 2828 (Internet Security Glossary) defines the Extensible Authentication Protocol as a framework that supports multiple, optional authentication mechanisms for PPP, including cleartext passwords, challenge-response, and arbitrary dialog sequences. It is intended for use primarily by a host or router that connects to a PPP network server via switched circuits or dial-up lines. The Remote Authentication Dial-In User Service (RADIUS) is defined as an Internet protocol for carrying dial-in user's authentication information and configuration information between a shared, centralized authentication server and a network access server that needs to authenticate the users of its network access ports. The other option is a distracter.

**SAML -** is an XML-based open standard data format for exchanging authentication and authorization data between parties, in particular, between an identity provider and a service provider. SAML is a product of the OASIS Security Services Technical Committee. SAML dates from 2001; the most recent update of SAML is from 2005.

The single most important problem that SAML addresses is the web browser single sign-on (SSO) problem. Single sign-on solutions are abundant at the intranet level (using cookies, for example) but extending these solutions beyond the intranet has been problematic and has led to the proliferation of non-interoperable proprietary technologies. (Another more recent approach to addressing the browser SSO problem is the OpenID protocol.)

The SAML specification defines three roles: the principal (typically a user), the identity provider (aka IdP), and the service provider (aka SP). In the use case addressed by SAML, the principal requests a service from the service provider. The service provider requests and obtains an identity assertion from the identity provider. On the basis of this assertion, the service provider can make an access control decision - in other words it can decide whether to perform some service for the connected principal.

Before delivering the identity assertion to the SP, the IdP may request some information from the principal - such as a user name and password - in order to authenticate the principal. SAML specifies the assertions between the three parties: in particular, the messages that assert identity that are passed from the IdP to the SP. In SAML, one identity provider may provide SAML assertions to many service providers. Conversely, one SP may rely on and trust assertions from many independent IdPs.

SAML does not specify the method of authentication at the identity provider; it may use a username/password, multifactor authentication, etc. A directory service, which allows users to login with a user name and password, is a typical source of authentication tokens (i.e., passwords) at an identity provider. Any of the popular common internet social services also provide identity services that in theory could be used to support SAML exchanges.

**XACML -** (extensible access control markup language) is a standard for **attribute-based access control**. XACML 3.0. The standard defines a declarative access control policy language implemented in XML and a processing model describing how to evaluate authorization requests according to the rules defined in policies.

As a published standard specification, one of the goals of XACML is to promote common terminology and interoperability between authorization implementations by multiple vendors. XACML is primarily an Attribute Based Access Control system (ABAC), where attributes (bits of data) associated with a user or action or resource are inputs into the decision of whether a given user may access a given resource in a particular way. Role-based access control (RBAC) can also be implemented in XACML as a specialization of ABAC.

The XACML model supports and encourages the separation of the authorization decision from the point of use. When authorization decisions are baked into client applications (or based on local machine userids and Access Control Lists (ACLs)), it is very difficult to update the decision criteria when the governing policy changes. When the client is decoupled from the authorization decision, authorization policies can be updated on the fly and affect all clients immediately.

XACML is structured into 3 levels of elements:

PolicySet,

Policy, and

Rule.

Both Rules and Requests use Subjects, Resources and Actions.

A Subject element is the entity requesting access. A Subject has one or more Attributes.

The Resource element is a data, service or system component. A Resource has a single Attribute.

An Action element defines the type of access requested on the Resource. Actions have one or more Attributes.

An Environment element can optionally provide additional information.

**LDAP (x.500)**

Bind - When an LDAP session is created, that is, when an LDAP client connects to the server, the authentication state of the session is set to anonymous. The BIND operation establishes the authentication state for a session.

Simple BIND and SASL PLAIN can send the user's DN and password in plaintext, so the connections utilizing either Simple or SASL PLAIN should be encrypted using Transport Layer Security (TLS). The server typically checks the password against the userPassword attribute in the named entry. Anonymous BIND (with empty DN and password) resets the connection to anonymous state.

SASL (Simple Authentication and Security Layer) BIND provides authentication services through a wide range of mechanisms, e.g. Kerberos or the client certificate sent with TLS.[8]

BIND also sets the LDAP protocol version. The version is an integer and at present <when?> must be either 2 (two) or 3 (three), although the standard supports integers between 1 and 127 (inclusive) in the protocol. If the client requests a version that the server does not support, the server must set the result code in the BIND response to the code for a protocol error. Normally clients should use LDAPv3, which is the default in the protocol but not always in LDAP libraries.

BIND had to be the first operation in a session in LDAPv2, but is not required in LDAPv3 (the current LDAP version). In LDAPv3, each successful BIND request changes the authentication state of the session and each unsuccessful BIND request resets the authentication state of the session.

**OpenID -** open standard that allows users to be authenticated by certain co-operating sites (known as Relying Parties or RP) using a third party service, eliminating the need for webmasters to provide their own ad hoc systems and allowing users to consolidate their digital identities.[1]

Users may create accounts with their preferred OpenID identity providers, and then use those accounts as the basis for signing on to any website which accepts OpenID authentication. The OpenID standard provides a framework for the communication that must take place between the identity provider and the OpenID acceptor (the "relying party").[2] An extension to the standard (the OpenID Attribute Exchange) facilitates the transfer of user attributes, such as name and gender, from the OpenID identity provider to the relying party (each relying party may request a different set of attributes, depending on its requirements).[3]

The OpenID protocol does not rely on a central authority to authenticate a user's identity. Moreover, neither services nor the OpenID standard may mandate a specific means by which to authenticate users, allowing for approaches ranging from the common (such as passwords) to the novel (such as smart cards or biometrics).

The term OpenID may also refer to an identifier as specified in the OpenID standard; these identifiers take the form of a unique URI, and are managed by some 'OpenID provider' that handles authentication.[1]

**CHAP**

A protocol that uses a three way hanbdshake The server sends the client a challenge which includes a random value(a nonce) to thwart replay attacks. The client responds with the MD5 hash of the nonce and the password.

CHAP authenticates a user or network host to an authenticating entity. That entity may be, for example, an Internet service provider. CHAP is specified in RFC 1994.

CHAP provides protection against replay attacks by the peer through the use of an incrementally changing identifier and of a variable challenge-value. CHAP requires that both the client and server know the plaintext of the secret, although it is never sent over the network. The MS-CHAP variant does not require either peer to know the plaintext, but has been broken.[1] Thus, CHAP provides better security as compared to Password Authentication Protocol (PAP).

CHAP is an authentication scheme used by Point to Point Protocol (PPP) servers to validate the identity of remote clients. CHAP periodically verifies the identity of the client by using a three-way handshake. This happens at the time of establishing the initial link (LCP), and may happen again at any time afterwards. The verification is based on a shared secret (such as the client user's password).[2]

After the completion of the link establishment phase, the authenticator sends a "challenge" message to the peer.

The peer responds with a value calculated using a one-way hash function on the challenge and the secret combined.

The authenticator checks the response against its own calculation of the expected hash value. If the values match, the authenticator acknowledges the authentication; otherwise it should terminate the connection.

At random intervals the authenticator sends a new challenge to the peer and repeats steps 1 through 3.

**Diameter**

Diameter is a protocol that has been developed to build upon the functionality of RADIUS and overcome many of its limitations. The creator of this protocol decided to call it Diameter as a play on the term RADIUS, as in the diameter is twice the radius.

Diameter is another AAA protocol that provides the same type of functionality as RADIUS and TACACS+ but also provides more flexibility and capabilities to meet the new demands of today’s complex and diverse networks where we want our wireless devices and smart phones to be able to authenticate themselves to our networks and we use roaming protocols, Mobile IP, PPPoE and etc.

Diameter provides a base protocol, which defines header formats, security options, commands, and AVPs (Attribute Value Pairs). This base protocol allows for extensions to tie in other services, such as VoIP, FoIP, Mobile IP, wireless, and cell phone authentication. So Diameter can be used as an AAA protocol for all of these different uses.

RADIUS and TACACS+ are client/server protocols, which mean that the server portion cannot send unsolicited commands to the client portion. The server portion can only speak when spoken to. Diameter is a peer-based protocol that allows either end to initiate communication.

This functionality allows the Diameter server to send a message to the access server to request the user to provide another authentication credential if she is attempting to access a secure resource.

This functionality also allows the Diameter server to disconnect the user if necessary for one reason or another.

Diameter is backward compatible with RADIUS, uses UDP and AVPs, and provides proxy server support.

It has better error detection and correction functionality and failover properties than RADIUS, thus provides better network resilience.

Diameter also provides end-to-end security through the use of IPSec or TLS, which is not available in RADIUS.

Diameter has the functionality and ability to provide the AAA functionality for other protocols and services because it has a large AVP set. RADIUS has 28 (256) AVPs and Diameter has 232. So, more AVPs allow for more functionality and services to exist and communicate between systems.

Diameter provides the following AAA function

Authentication

PAP, CHAP, EAP

End-to-end protection of authentication information

Replay attack protection

Authorization

Redirects, secure proxies, relays, and brokers

State reconciliation

Unsolicited disconnect

Reauthorization on demand

Accounting

Reporting, ROAMOPS accounting, event monitoring

**OAuth**

is an open standard for authorization. OAuth provides a method for clients to access server resources on behalf of a resource owner (such as a different client or an end-user). It also provides a process for end-users to authorize third-party access to their server resources without sharing their credentials (typically, a username and password pair), using user-agent redirections.

OAuth is a service that is complementary to, and therefore distinct from, OpenID. OAuth is also distinct from OATH, which is a reference architecture for authentication (i.e. not a standard).

**OpenID vs. pseudo-authentication using OAuth**

The following diagrams highlight the differences between using OpenID and OAuth for authentication. With OpenID, the process starts with the application asking the user for their identity (basically a log-in request by the application, to which the user typically provides an OpenID URI rather than actual credentials). In the case of OAuth, the application specifically requests a limited access OAuth Token (valet key) to access the APIs (enter the house) on the user's behalf (which typically explicitly names the particular rights requested, and does not require the user to enter credentials at all). If the user can grant that access, the application can retrieve the unique identifier for establishing the profile (identity) using the APIs. In either case, the access to the Identity Provider will involve authentication to the Identity Provider, unless some session is already in effect. The result in the OpenID case is that the application allows the user access, because it trusts the OpenID Identity provider. The result in the OAuth case is that the API provider allows the application access because it trusts its own valet keys.

**SINGLE SIGN ON:**

The advantages of SSO include having the ability to use stronger passwords, easier administration as far as changing or deleting the passwords, minimize the risks of orphan accounts, and requiring less time to access resources.

SSO can be implemented by using scripts that replay the users multiple log-ins, or by using authentication servers to verify a user's identity and encrypted authentication tickets to permit access to system services.

Single Sign on was the best answer in this case because it would include Kerberos.

**Benefits of using single sign-on include:**

Reducing password fatigue from different user name and password combinations

Reducing time spent re-entering passwords for the same identity

Reducing IT costs due to lower number of IT help desk calls about passwords

SSO shares centralized authentication servers that all other applications and systems use for authentication purposes and combines this with techniques to ensure that users do not have to actively enter their credentials more than once.

SSO users do not need to remember so many passwords to log in to different systems or applications.

**Drawbacks:**

The term enterprise reduced sign-on is preferred by some authors[who?] who believe single sign-on to be impossible in real use cases.

As single sign-on provides access to many resources once the user is initially authenticated ("keys to the castle") it increases the negative impact in case the credentials are available to other persons and misused. Therefore, single sign-on requires an increased focus on the protection of the user credentials, and should ideally be combined with strong authentication methods like smart cards and one-time password tokens.

Single sign-on also makes the authentication systems highly critical; a loss of their availability can result in denial of access to all systems unified under the SSO. SSO can thus be undesirable for systems to which access must be guaranteed at all times, such as security or plant-floor systems

**OTHERS:**

**SmartCards -** Tamper-resistant microprocessors are used to store and process private or sensitive information, such as private keys or electronic money credit. To prevent an attacker from retrieving or modifying the information, the chips are designed so that the information is not accessible through external means and can be accessed only by the embedded software, which should contain the appropriate security measures.

The main difference between memory cards and smart cards is their capacity to process information. A memory card holds information but cannot process information. A smart card holds information and has the necessary hardware and software to actually process that information.

**A memory card** holds a user’s authentication information, so that this user needs only type in a user ID or PIN and presents the memory card to the system. If the entered information and the stored information match and are approved by an authentication service, the user is successfully authenticated.

A common example of a memory card is a swipe card used to provide entry to a building. The user enters a PIN and swipes the memory card through a card reader. If this is the correct combination, the reader flashes green and the individual can open the door and enter the building.

Memory cards can also be used with computers, but they require a reader to process the information. The reader adds cost to the process, especially when one is needed for every computer. Additionally, the overhead of PIN and card generation adds additional overhead and complexity to the whole authentication process. However, a memory card provides a more secure authentication method than using only a password because the attacker would need to obtain the card and know the correct PIN.

Administrators and management need to weigh the costs and benefits of a memory card implementation as well as the security needs of the organization to determine if it is the right authentication mechanism for their environment.

One of the most prevalent weaknesses of memory cards is that data stored on the card are not protected. Unencrypted data on the card (or stored on the magnetic strip) can be extracted or copied. Unlike a smart card, where security controls and logic are embedded in the integrated circuit, memory cards do not employ an inherent mechanism to protect the data from exposure.

**Hybrid Card**

A hybrid card has two chips, one with a contact interface and one with a contactless interface. **The two chips are not interconnected.**

**Dual-Interface card**

Do not confuse this card with the Hybrid Card. This one has only one chip. A dual-interface card has a single chip with both contact and contactless interfaces. With dual-interface cards, it is possible to access the same chip using either a contact or contactless interface with a very high level of security.

**Inner working of the cards**

The chips used in all of these cards fall into two categories as well: microcontroller chips and memory chips. A memory chip is like a small floppy disk with optional security. Memory chips are less expensive than microcontrollers but with a corresponding decrease in data management security. Cards that use memory chips depend on the security of the card reader for processing and are ideal for situations that require low or medium security.

A microcontroller chip can add, delete, and otherwise manipulate information in its memory. A microcontroller is like a miniature computer, with an input/output port, operating system, and hard disk. Smart cards with an embedded microcontroller have the unique ability to store large amounts of data, carry out their own on-card functions (e.g., encryption and digital signatures) and interact intelligently with a smart card reader.

**The selection of a particular card technology is driven by a variety of issues, including:**

Application dynamics

Prevailing market infrastructure

Economics of the business model

Strategy for shared application cards

**Smart cards are used in many applications worldwide, including:**

Secure identity applications - employee ID badges, citizen ID documents, electronic passports, driver’s licenses, online authentication devices

Healthcare applications - citizen health ID cards, physician ID cards, portable medical records cards

Payment applications - contact and contactless credit/debit cards, transit payment cards

Telecommunications applications - GSM Subscriber Identity Modules, pay telephone payment cards

**Detective/administrative -** Additional detective/administrative controls are job rotation, the sharing of responsibilities, and reviews of audit records.

**TECHNICAL CONTROLS**

Technical security involves the use of safeguards incorporated in computer hardware, operations or applications software, communications hardware and software, and related devices. Technical controls are sometimes referred to as logical controls.

Preventive Technical Controls

Preventive technical controls are used to prevent unauthorized personnel or programs from gaining remote access to computing resources. Examples of these controls include:

• Access control software.

• Antivirus software.

• Library control systems.

• Passwords.

• Smart cards.

• Encryption.

• Dial-up access control and callback systems.

Preventive Physical Controls

Preventive physical controls are employed to prevent unauthorized personnel from entering computing facilities (i.e., locations housing computing resources, supporting utilities, computer hard copy, and input data media) and to help protect against natural disasters. Examples of these controls include:

• Backup files and documentation.

• Fences.

• Security guards.

• Badge systems.

• Double door systems.

• Locks and keys.

• Backup power.

• Biometric access controls.

• Site selection.

• Fire extinguishers.

Preventive Administrative Controls

Preventive administrative controls are personnel-oriented techniques for controlling people’s behavior to ensure the confidentiality, integrity, and availability of computing data and programs. Examples of preventive administrative controls include:

• Security awareness and technical training.

• Separation of duties.

• Procedures for recruiting and terminating employees.

• Security policies and procedures.

• Supervision.

• Disaster recovery, contingency, and emergency plans.

• User registration for computer access.

**Security awareness** is a **preventive measure**, not a compensating measure for access violations.

**Preventive controls** are concerned with avoiding **occurrences of risks while** deterrent controls are concerned with discouraging violations. Detecting controls identify occurrences and compensating controls are alternative controls, used to compensate weaknesses in other controls. Supervision is an example of compensating control.

**Access control lists** are object-based whereas **capability tables** are subject-based.

Capability tables are used to track, manage and apply controls based on the object and rights, or capabilities of a subject. For example, a table identifies the object, specifies access rights allowed for a subject, and permits access based on the user's posession of a capability (or ticket) for the object. It is a row within the matrix.

To put it another way, A capabiltiy table is different from an ACL because the subject is bound to the capability table, whereas the object is bound to the ACL.

It [ACL] specifies a list of users [subjects] who are allowed access to each object" CBK, p. 188

A capability table is incorrect. "Capability tables are used to track, manage and apply controls based on the object and rights, or capabilities of a subject. For example, a table identifies the object, specifies access rights allowed for a subject, and permits access based on the user's posession of a capability (or ticket) for the object." CBK, pp. 191-192. The distinction that makes this an incorrect choice is that access is based on posession of a capability by the subject.

To put it another way. "A capabiltiy table is different from an ACL because the subject is bound to the capability table, whereas the object is bound to the ACL."

An access control matrix is incorrect.

**The access control matrix** is a way of describing the rules for an access control strategy. The matrix lists the users, groups and roles down the left side and the resources and functions across the top. The cells of the matrix can either indicate that access is allowed or indicate the type of access. Harris describes it as a table if subjects and objects specifying the access rights a certain subject possesses pertaining to specific objects.

In either case, the matrix is a way of analyzing the access control needed by a population of subjects to a population of objects. This access control can be applied using rules, ACL's, capability tables, etc.

Access Matrix Model

An access control matrix is a table of subjects and objects indicating what actions individual subjects can take upon individual objects. Matrices are data structures that programmers implement as table lookups that will be used and enforced by the operating system.

This type of access control is usually an attribute of DAC models. The access rights can be assigned directly to the subjects (capabilities) or to the objects (ACLs).

**Honeypots -** The primary purpose of a honeypot is to study the attack methods of an attacker for the purposes of understanding their methods and improving defenses.

**Environmental error,** the environment in which a system is installed somehow causes the system to be vulnerable. This may be due, for example, to an unexpected interaction between an application and the operating system or between two applications on the same host. A configuration error occurs when user controllable settings in a system are set such that the system is vulnerable. In an access validation error, the system is vulnerable because the access control mechanism is faulty. In an exceptional condition handling error, the system somehow becomes vulnerable due to an exceptional condition that has arisen.

**A network-based vulnerability assessment system** either re-enacts system attacks, noting and recording responses to the attacks, or probes different targets to infer weaknesses from their responses. Since they are actively attacking or scanning targeted systems, network-based vulnerability assessment systems are also **called active vulnerability systems**. Credential-based and passive are related to host-based vulnerability assessment systems.

**Cognitive passwords** are fact or opinion-based information used to verify an individual's identity. Passwords that can be used only once are one-time or dynamic passwords. Password generators that use a challenge response scheme refer to token devices.

A passphrase is a sequence of characters that is longer than a password and is transformed into a virtual password.

**Reporting structure for the Information Systems Security function** - Operations would rank lowest of the available answers as they are more apt to be management so would be considered least effective structure for the Information Systems Security function.

In order to offer more independence and get more attention from management, an IT/IS security function should be independent from IT/IS operations and ideally report directly to the CEO. If it were to report to IT/IS, operations is probably the last function the IS Security function should be reporting to.

**DB VIEW**

A number of operations can be performed in a relational algebra to build relations and operate on the data. Five operations are primitives (Select, Project, Union, Difference and Product) and the other operations can be defined in terms of those five. A View is defined from the operations of Join, Project, and Select

External consistency ensures that the data stored in the database is consistent with the real world.

**INSIDE ATTACK -** An inside attack is an attack initiated by an entity inside the security perimeter, an entity that is authorized to access system resources but uses them in a way not approved by those who granted the authorization whereas an outside attack is initiated from outside the perimeter, by an unauthorized or illegitimate user of the system. An active attack attempts to alter system resources to affect their operation and a passive attack attempts to learn or make use of the information from the system but does not affect system resources.

**PASSWORD TOKENS:**

Tokens are electronic devices or cards that supply a user's password for them. A token system can be used to supply either a static or a dynamic password. There is a big difference between the static and dynamic systems, a static system will normally log a user in but a dynamic system the user will often have to log themselves in.

Static Password Tokens:

The owner identity is authenticated by the token. This is done by the person who issues the token to the owner (normally the employer). The owner of the token is now authenticated by "something you have". The token authenticates the identity of the owner to the information system. An example of this occurring is when an employee swipes his or her smart card over an electronic lock to gain access to a store room.

**Synchronous Dynamic Password Tokens:**

This system is a lot more complex then the static token password. The synchronous dynamic password tokens generate new passwords at certain time intervals that are synched with the main system. The password is generated on a small device similar to a pager or a calculator that can often be attached to the user's key ring. Each password is only valid for a certain time period, typing in the wrong password in the wrong time period will invalidate the authentication. The time factor can also be the systems downfall. If a clock on the system or the password token device becomes out of synch, a user can have troubles authenticating themselves to the system.

**- The token generates a new unique password value at fixed time intervals (this password could be the time of day encrypted with a secret key).**

**- the unique password is entered into a system or workstation along with an owner's PIN.**

**- The authentication entity in a system or workstation knows an owner's secret key and PIN, and the entity verifies that the entered password is valid and that it was entered during the valid time window.**

**Synchronous Tokens:**

A Synchronous token generates a one-time password that is only valid for a short period of time. Once the password is used it is no longer valid, and it expires if not entered in the acceptable time frame.

**Asynchronous Dynamic Password Tokens:**

The clock synching problem is eliminated with asynchronous dynamic password tokens. This system works on the same principal as the synchronous one but it does not have a time frame. A lot of big companies use this system especially for employee's who may work from home on the companies VPN (Virtual private Network).

Challenge Response Tokens:

This is an interesting system. A user will be sent special "challenge" strings at either random or timed intervals. The user inputs this challenge string into their token device and the device will respond by generating a challenge response. The user then types this response into the system and if it is correct they are authenticated.

**BUSINESS AVAILABILITY PLANNING:**

The following measures are used to compensate for both internal and external access violations:

- Backups

- RAID (Redundant Array of Independent Disks) technology

- Fault tolerance

- Business Continuity Planning

- Insurance

**ASSURANCE PROCEDURES:**

Controls provide accountability for individuals accessing information. Assurance procedures ensure that access control mechanisms correctly implement the security policy for the entire life cycle of an information system.

**Information Flow Model combined with Bell Lapadula**

Securing the data manipulated by computing systems has been a challenge in the past years. Several methods to limit the information disclosure exist today, such as access control lists, firewalls, and cryptography. However, although these methods do impose limits on the information that is released by a system, they provide no guarantees about information propagation. For example, access control lists of file systems prevent unauthorized file access, but they do not control how the data is used afterwards. Similarly, cryptography provides a means to exchange information privately across a non-secure channel, but no guarantees about the confidentiality of the data are given once it is decrypted.

In low level information flow analysis, each variable is usually assigned a security level. The basic model comprises two distinct levels: low and high, meaning, respectively, publicly observable information, and secret information. To ensure confidentiality, flowing information from high to low variables should not be allowed. On the other hand, to ensure integrity, flows to high variables should be restricted.

More generally, the security levels can be viewed as a lattice with information flowing only upwards in the lattice.

**Noninterference Models**

This could have been another good answer as it would help in minimizing the damage from covert channels.

The goal of a noninterference model is to help ensure that high-level actions (inputs) do not determine what low-level user s can see (outputs ) . Most of the security models presented are secured by permitting restricted ﬂows between high- and low-level users. The noninterference model maintains activities at different security levels to separate these levels from each other. In this way, it minimizes leakages that may happen through covert channels, because there is complete separation (noninterference) between security levels. Because a user at a higher security level has no way to interfere with the activities at a lower level, the lower-level user cannot get any information from the higher leve.

**TAKE-GRANT SYSTEM models** a protection system which consists of a set of states and state transitions. A directed graph shows the connections between the nodes of this system. These nodes are representative of the subjects or objects of the model. The directed edges between the nodes represent the rights that one node has over the linked node.

**Port knocking -** is where the client will attempt to connect to a predefined set of ports to identify him as an authorized client. The port knocking sequence is used to identify the client as a legitimate user.

**Cryptcat -** Twofish encryption to encrypt network traffic thereby evading IDS/IDP detection.

Netcat is a utility that can be used to open ports on a compromised host.

Cryptcat does this but supports twofish (Schneier) encryption which is not decryptable by an IDS in transit.

**IDS/IPS**

Intrusion Detection attempts to identify and isolate computer and network attacks by observing network logs or other audit data.

Audit trails maintain a record of system activity both by system and application processes and by user activity of systems and applications. In conjunction with appropriate tools and procedures, audit trails can assist in detecting security violations, performance problems, and flaws in applications.

Intrusion detection refers to the process of identifying attempts to penetrate a system and gain unauthorized access. If audit trails have been designed and implemented to record appropriate information, they can assist in intrusion detection. Although normally thought of as a real-time effort, intrusions can be detected in real time, by examining audit records as they are created (or through the use of other kinds of warning flags/notices), or after the fact (e.g., by examining audit records in a batch process).

Real-time intrusion detection is primarily aimed at outsiders attempting to gain unauthorized access to the system. It may also be used to detect changes in the system's performance indicative of, for example, a virus or worm attack (forms of malicious code). There may be difficulties in implementing real-time auditing, including unacceptable system performance.

After-the-fact identification may indicate that unauthorized access was attempted (or was successful). Attention can then be given to damage assessment or reviewing controls that were attacked.

**Statistical Anomaly-Based IDS - With** this method, an IDS acquires data and defines a "normal" usage profile for the network or host that is being monitored. Some disadvantages of a statistical anomaly-based ID are that it will not detect an attack that does not significantly change the system operating characteristics, or it may falsely detect a non-attack event that had caused a momentary anomaly in the system.

**A network-based IDS is passive while it acquires data.**

**Wardialling countermeasure** - Knowledge of modem numbers is a poor access control method as an attacker can discover modem numbers by dialing all numbers in a range. Requiring user authentication before remote access is granted will help in avoiding unauthorized access over a modem line.

**LOGO BANNERS (external users) VS WRITTEN AGREEMENT (internal users)**

There are two possible answers based on how the question is presented, this question could either apply to internal users or ANY anonymous/external users.

Internal users should always have a written agreement first, then logon banners serve as a constant reminder.

Banners at the log-on time should be used to notify external users of any monitoring that is being conducted. A good banner will give you a better legal stand and also makes it obvious the user was warned about who should access the system, who is authorized and unauthorized, and if it is an unauthorized user then he is fully aware of trespassing. Anonymous/External users, such as those logging into a web site, ftp server or even a mail server; their only notification system is the use of a logon banner.

**Incident handling:**

**Analysis and tracking.** In this step, your main objective is to examine and analyze what has occurred and focus on determining the root cause of the incident.

**Recovery** is incorrect as recovery is about resuming operations or bringing affected systems back into production

**Containment** is incorrect as containment is about reducing the potential impact of an incident.

**Triage** is incorrect as triage is about determining the seriousness of the incident and filtering out false positives

**Access Control Objectives:**

Availability assures that a system's authorized users have timely and uninterrupted access to the information in the system. **The additional access control objectives are reliability and utility.** These and other related objectives flow from the organizational security policy. This policy is a high-level statement of management intent regarding the control of access to information and the personnel who are authorized to receive that information. Three things that must be considered for the planning and implementation of access control mechanisms are the threats to the system, the system's vulnerability to these threats, and the risk that the threat may materialize

**Batch and Scripts**

Because scripts contain credentials, they must be stored in a protected area and the transmission of the scripts must be dealt with carefully. Operators might need access to batch files and scripts. The least privilege concept requires that each subject in a system be granted the most restrictive set of privileges needed for the performance of authorized tasks. The need-to-know principle requires a user having necessity for access to, knowledge of, or possession of specific information required to perform official tasks or services.

**Constrained user interfaces** limit the functions that can be selected by a user.

**Emanation attacks** are the act of intercepting electrical signals that radiate from computing equipment. There are several countermeasures including shielding cabling, white noise, control zones, and TEMPEST equipment (this is a Faraday cage around the equipment)

**LC and MAC; IEEE 802.2 and 802.3**

The data link layer, or Layer 2, of the OSI model is responsible for adding a header and a trailer to a packet to prepare the packet for the local area network or wide area network technology binary format for proper line transmission.

Layer 2 is divided into two functional sublayers.

The upper sublayer is the Logical Link Control (LLC) and is defined in the IEEE 802.2 specification. It communicates with the network layer, which is immediately above the data link layer.

Below the LLC is the Media Access Control (MAC) sublayer, which specifies the interface with the protocol requirements of the physical layer.

Thus, the specification for this layer depends on the technology of the physical layer.

The IEEE MAC specification for Ethernet is 802.3, Token Ring is 802.5, wireless LAN is 802.11, and so on. When you see a reference to an IEEE standard, such as 802.11 or 802.16, it refers to the protocol working at the MAC sublayer of the data link layer of the protocol stack.