**4.A Identify security architecture approach**

**4.A.1 Types and scope (e.g. enterprise, network, SOA)**

**4.A.2 Frameworks**

**Sherwood Applied Business Security Architecture - SABSA**

SABSA is a model and a methodology for developing risk-driven enterprise information security architectures and for delivering security infrastructure solutions that support critical business initiatives. The primary characteristic of the SABSA model is that everything must be derived from an analysis of the business requirements for security, especially those in which security has an enabling function through which new business opportunities can be developed and exploited.

The process analyzes the business requirements at the outset, and creates a chain of traceability through the strategy and concept, design, implementation, and ongoing ‘manage and measure’ phases of the lifecycle to ensure that the business mandate is preserved. Framework tools created from practical experience further support the whole methodology.

The model is layered, with the top layer being the business requirements definition stage. At each lower layer a new level of abstraction and detail is developed, going through the definition of the conceptual architecture, logical services architecture, physical infrastructure architecture and finally at the lowest layer, the selection of technologies and products (component architecture).

The SABSA model itself is generic and can be the starting point for any organization, but by going through the process of analysis and decision-making implied by its structure, it becomes specific to the enterprise, and is finally highly customized to a unique business model. It becomes in reality the enterprise security architecture, and it is central to the success of a strategic program of information security management within the organization.

SABSA is a particular example of a methodology that can be used both for IT (Information Technology) and OT (Operational Technology) environments.

**SOMF**

The service-oriented modeling framework (SOMF) has been proposed by author Michael Bell as a holistic and anthropomorphic modeling language for software development that employs disciplines and a universal language to provide tactical and strategic solutions to enterprise problems.[2] The term "holistic language" pertains to a modeling language that can be employed to design any application, business and technological environment, either local or distributed. This universality may include design of application-level and enterprise-level solutions, including SOA landscapes or cloud computing environments. The term "anthropomorphic", on the other hand, affiliates the SOMF language with intuitiveness of implementation and simplicity of usage. Furthermore, The SOMF language and its notation has been adopted by Sparx Enterprise Architect modeling platform that enables business architects, technical architects, managers, modelers, developers, and business and technical analysts to pursue the chief SOMF life cycle disciplines.

SOMF is a service-oriented development life cycle methodology, a discipline-specific modeling process. It offers a number of modeling practices and disciplines that contribute to a successful service-oriented life cycle development and modeling during a project (see image on left).

**Zachman framework**

The Zachman Framework is an Enterprise Architecture framework for enterprise architecture, which provides a formal and highly structured way of viewing and defining an enterprise. It consists of a two dimensional classification matrix based on the intersection of six communication questions (What, Where, When, Why, Who and How) with six levels of reification, successively transforming the abstract ideas on the Scope level into concrete instantiations of those ideas at the Operations level.[1]

The Zachman Framework is a schema for organizing architectural artefacts (in other words, design documents, specifications, and models) that takes into account both whom the artefact targets (for example, business owner and builder) and what particular issue (for example, data and functionality) is being addressed.[2] The Zachman Framework is not a methodology in that it does not imply any specific method or process for collecting, managing, or using the information that it describes.[3]

The Framework is named after its creator John Zachman, who first developed the concept in the 1980s at IBM. It has been updated several times since.[4]

The term "Zachman Framework" has multiple meanings. It can refer to any of the frameworks proposed by John Zachman:

The initial framework, named A Framework for Information Systems Architecture, by John Zachman published in an 1987 article in the IBM Systems journal.[5]

The Zachman Framework for Enterprise Architecture, an update of the 1987 original in the 1990s extended and renamed .[6]

One of the later versions of the Zachman Framework, offered by Zachman International as industry standard.

Collage of Zachman Frameworks as presented in several books on Enterprise Architecture from 1997 to 2005.

In other sources the Zachman Framework is introduced as a framework, originated by and named after John Zachman, represented in numerous ways, see image. This framework is explained as, for example:

a framework to organize and analyze data,[7]

a framework for enterprise architecture.[8]

a classification system, or classification scheme[9]

a matrix, often in a 6x6 matrix format

a two-dimensional model[10] or an analytic model.

a two-dimensional schema, used to organize the detailed representations of the enterprise.[11]

Beside the frameworks developed by John Zachman numerous extensions and or applications have been developed, which are also sometimes called Zachman Frameworks.

The Zachman Framework summarizes a collection of perspectives involved in enterprise architecture. These perspectives are represented in a two-dimensional matrix that defines along the rows the type of stakeholders and with the columns the aspects of the architecture. The framework does not define a methodology for an architecture. Rather, the matrix is a template that must be filled in by the goals/rules, processes, material, roles, locations, and events specifically required by the organization. Further modeling by mapping between columns in the framework identifies gaps in the documented state of the organization.[12]

The framework is a simple and logical structure for classifying and organizing the descriptive representations of an enterprise. It is significant to both the management of the enterprise, and the actors involved in the development of enterprise systems.[13] While there is no order of priority for the columns of the Framework, the top-down order of the rows is significant to the alignment of business concepts and the actual physical enterprise. The level of detail in the Framework is a function of each cell (and not the rows). When done by IT the lower level of focus is on information technology, however it can apply equally to physical material (ball valves, piping, transformers, fuse boxes for example) and the associated physical processes, roles, locations etc. related to those items.

**Orange Book**

**The Trusted Computer System Evaluation Criteria (TCSEC) = ORANGE BOOK**

**Focused on Confidentiality NOT integrity.**

One of the commonly cited deficiencies of the Orange Book is that it addresses **only the operating system** and does not include networks, databases, etc.

According to the Orange book, Assurance refers to **Operational Assurance**, such as system architecture, protected execution domain, system integrity; and **Life-Cycle Assurance**, such as design methodology, security testing, and configuration.

Purpose:

* To provide a standard methodology for measuring the amount of trust you can place in a system
* To produce standards as to what security features for manufacturers to include when developing new commercial products.
* To provide government user personnel with a basis for the specific security requirements they were demanding when purchasing products.

The Bell-LaPadula model is the security policy model on which the Orange Book requirements are based. From the Orange Book definition, "A formal state transition model of computer security policy that describes a set of access control rules. In this formal model, the entities in a computer system are divided into abstract sets of subjects and objects. The notion of secure state is defined and it is proven that each state transition preserves security by moving from secure state to secure state; thus, inductively proving the system is secure. A system state is defined to be 'secure' if the only permitted access modes of subjects to objects are in accordance with a specific security policy. In order to determine whether or not a specific access mode is allowed, the clearance of a subject is compared to the classification of the object and a determination is made as to whether the subject is authorized for the specific access mode."

The Orange book requires protection against two types of **covert channels**, **Timing and Storage.**

**Covert Storage Channel:** A covert channel that involves writing to a storage location by one process and the direct or indirect reading of the storage location by another process. Covert storage channels typically involve a resource (for example, sectors on a disk) that is shared by two subjects at different security levels.

**Covert Timing Channel:** A covert channel in which one process modulates its system resource (for example, CPU cycles), which is interpreted by a second process as some type of communication.

The **TCSEC**, frequently referred to as the Orange Book, is the centerpiece of the DoD Rainbow Series publications.

Initially issued by the National Computer Security Center (NCSC) an arm of the National Security Agency in 1983 and then updated in 1985, TCSEC was replaced with the development of the Common Criteria international standard originally published in 2005.

**D – Minimal protection**

**C – Discretionary protection**

**C1 – Discretionary Security Protection**

* C1 involves discretionary protection, C2 involves controlled access protection, B1 involves labeled security protection and B2 involves structured protection.

**C2 – Controlled Access Protection**

* Magnetic media must not have any remanence of previous data in order to be reused. This also applies to buffers, cache and other memory allocation and is required at TCSEC C2 levels.

**B – Mandatory Protection**

**B1 – Labeled Security**

* Level B1 would be most similar to Compartmented Mode Workstations (CMW). The CMW has many of the requirements of B1 with some of the requirements listed under B2 and B3 as well. It fits the best with B1 as a level.
* B1 is also called "Labeled Security" and each data object must have a classification label and each subject a clearence label. On each access attempt, the classification and clearence are checked to verify that the access is permissible.
* Level B1 is the first to require design specification and verification and this would also be a requirement for all higher levels.

**B2 – Structured Protection**

* the system must protect against covert storage channels.This class ("Structured Protection") requires more stringent authentication mechanisms and well-defined interfaces between layers. Subjects and devices require labels and the system must not allow covert channels.
* B2 security level requires that systems must support separate operator and system administrator roles.
* Level B2 is the first level to require configuration management and this would also be a requirement for all higher levels.
* Trusted facility management is an assurance requirement only for highly secure systems (B2, B3 and A1), but many systems evaluated a lower security levels are structured to try to meet this requirement.
* Configuration management is the process of tracking and approving changes to a system. It is only required for B2, B3 and A1 level systems, but because it is common sense, it is also recommended for systems that are evaluated at lower levels.

**B3 – Security Domains**

* Level B3 is the first level requiring Security Domains.
* At B3 and A1, systems must clearly identify the functions of the security administrator to perform the security-related functions.
* Trusted facility management is an assurance requirement only for highly secure systems (B2, B3 and A1), but many systems evaluated a lower security levels are structured to try to meet this requirement.
* Configuration management is the process of tracking and approving changes to a system. It is only required for B2, B3 and A1 level systems, but because it is common sense, it is also recommended for systems that are evaluated at lower levels.

**A – Verified Protection**

**A1 – Verified Design**

* Level A1 is Verified Design, this ensures nothing has been tampered with, not even the documentation so Trusted Distribution would be required at this level.
* At B3 and A1, systems must clearly identify the functions of the security administrator to perform the security-related functions.
* Trusted facility management is an assurance requirement only for highly secure systems (B2, B3 and A1), but many systems evaluated a lower security levels are structured to try to meet this requirement.
* Configuration management is the process of tracking and approving changes to a system. It is only required for B2, B3 and A1 level systems, but because it is common sense, it is also recommended for systems that are evaluated at lower levels.

**Common Criteria**

**Assurance**

**ITSEC + Orange Book + CTCPEC (Canadian)**

Common Criteria evaluations are performed on computer security products and systems.

Protection profiles and security targets are elements of the **ISO International Standard 15408** "Evaluation Criteria for Information Technology Security", also commonly known as the Common Criteria (CC).ISO 15408 is also know as Common Criteria. Common Criteria terminology includes Protection Profiles and Security Target. This terminology was never used for ITSEC, only ToE was used so the correct answer would be **ISO 15408.**

* **Target Of Evaluation (TOE)** – the product or system that is the subject of the evaluation.

The evaluation serves to validate claims made about the target. To be of practical use, the evaluation must verify the target's security features. This is done through the following:

* [**Protection Profile**](https://en.wikipedia.org/wiki/Protection_Profile)**(PP)** – a document, typically created by a user or user community, which identifies security requirements for a class of security devices (for example, [smart cards](https://en.wikipedia.org/wiki/Smart_card) used to provide [digital signatures](https://en.wikipedia.org/wiki/Digital_signature), or network [firewalls](https://en.wikipedia.org/wiki/Firewall_(computing))) relevant to that user for a particular purpose. Product vendors can choose to implement products that comply with one or more PPs, and have their products evaluated against those PPs. In such a case, a PP may serve as a template for the product's ST (Security Target, as defined below), or the authors of the ST will at least ensure that all requirements in relevant PPs also appear in the target's ST document. Customers looking for particular types of products can focus on those certified against the PP that meets their requirements.
* [**Security Target**](https://en.wikipedia.org/wiki/Security_Target)**(ST)** – the document that identifies the security *properties* of the target of evaluation. It may refer to one or more PPs. The TOE is evaluated against the SFRs (see below) established in its ST, no more and no less. This allows vendors to tailor the evaluation to accurately match the intended capabilities of their product. This means that a network firewall does not have to meet the same functional requirements as a [database](https://en.wikipedia.org/wiki/Database) management system, and that different firewalls may in fact be evaluated against completely different lists of requirements. The ST is usually published so that potential customers may determine the specific security features that have been certified by the evaluation.
* **Security Functional Requirements (SFRs)** – specify individual security **functions** which may be provided by a product. The Common Criteria presents a standard catalogue of such functions. For example, a SFR may state **how** a user acting a particular [role](https://en.wikipedia.org/wiki/RBAC) might be [authenticated](https://en.wikipedia.org/wiki/Authentication). The list of SFRs can vary from one evaluation to the next, even if two targets are the same type of product. Although Common Criteria does not prescribe any SFRs to be included in an ST, it identifies dependencies where the correct operation of one function (such as the ability to limit access according to roles) is dependent on another (such as the ability to identify individual roles).

The evaluation process also tries to establish the level of confidence that may be placed in the product's security features through [quality assurance](https://en.wikipedia.org/wiki/Quality_assurance) processes:

* **Security Assurance Requirements (SARs)** – descriptions of the measures taken during development and evaluation of the product to assure compliance with the claimed security functionality. For example, an evaluation may require that all source code is kept in a change management system, or that full functional testing is performed. The Common Criteria provides a catalogue of these, and the requirements may vary from one evaluation to the next. The requirements for particular targets or types of products are documented in the ST and PP, respectively.
* [**Evaluation Assurance Level**](https://en.wikipedia.org/wiki/Evaluation_Assurance_Level)**(EAL)** – the numerical rating describing the depth and rigor of an evaluation. Each EAL corresponds to a package of security assurance requirements (SARs, see above) which covers the complete development of a product, with a given level of strictness. Common Criteria lists seven levels, with EAL 1 being the most basic (and therefore cheapest to implement and evaluate) and EAL 7 being the most stringent (and most expensive). Normally, an ST or PP author will not select assurance requirements individually but choose one of these packages, possibly 'augmenting' requirements in a few areas with requirements from a higher level. Higher EALs *do not* necessarily imply "better security", they only mean that the claimed security assurance of the TOE has been more extensively [verified](https://en.wikipedia.org/wiki/Verification_and_validation).

So far, most PPs and most evaluated STs/certified products have been for IT components (e.g., firewalls, [operating systems](https://en.wikipedia.org/wiki/Operating_system), smart cards). Common Criteria certification is sometimes specified for IT procurement. Other standards containing, e.g., interoperation, system management, user training, supplement CC and other product standards. Examples include the [ISO/IEC 17799](https://en.wikipedia.org/wiki/ISO/IEC_17799) (Or more properly BS 7799-1, which is now [ISO/IEC 27002](https://en.wikipedia.org/wiki/ISO/IEC_27002)) or the German [IT-Grundschutzhandbuch](https://de.wikipedia.org/wiki/IT-Grundschutz).

Details of cryptographic implementation within the TOE are outside the scope of the CC. Instead, national standards, like [FIPS 140-2](https://en.wikipedia.org/wiki/FIPS_140-2) give the specifications for cryptographic modules, and various standards specify the cryptographic algorithms in use.

More recently, PP authors are including cryptographic requirements for CC evaluations that would typically be covered by FIPS 140-2 evaluations, broadening the bounds of the CC through scheme-specific interpretations.

According to the Common Criteria, **an intermediate combination of security requirement** components is termed a package. The package permits the expression of a set of either functional or assurance requirements that meet some particular need, expressed as a set of security objectives. A package may be used in the construction of more complex packages or Protection Profiles and Security Targets. The seven evaluation assurance levels (EALs) are predefined assurance packages. The TOE is an IT product or system to be evaluated.

Under the Common Criteria model, an evaluation is carried out on a product and is assigned an Evaluation Assurance Level (EAL). The thorough and stringent testing increases in detailed-oriented tasks as the assurance levels increase. The Common Criteria has seven assurance levels. The range is from EAL1, where functionality testing takes place, to EAL7, where thorough testing is performed and the system design is verified.

The Orange Book and the Rainbow Series provide evaluation schemes that are too rigid and narrowly defined for the business world. ITSEC attempted to provide a more flexible approach by separating the functionality and assurance attributes and considering the evaluation of entire systems. However, this flexibility added complexity because evaluators could mix and match functionality and assurance ratings, which resulted in too many classifications to keep straight.

Because we are a species that continues to try to get it right, the next attempt for an effective and usable evaluation criteria was the Common Criteria. In 1990, the International Organization for Standardization (ISO) identified the need for international standard evaluation criteria to be used globally. The Common Criteria project started in 1993 when several organizations came together to combine and align existing and emerging evaluation criteria (TCSEC, ITSEC, Canadian Trusted Computer Product Evaluation Criteria [CTCPEC], and the Federal Criteria).

The Common Criteria was developed through a collaboration among national security standards organizations within the United States, Canada, France, Germany, the United Kingdom, and the Netherlands. The benefit of having a globally recognized and accepted set of criteria is that it helps consumers by reducing the complexity of the ratings and eliminating the need to understand the definition and meaning of different ratings within various evaluation schemes. This also helps vendors, because now they can build to one specific set of requirements if they want to sell their products internationally, instead of having to meet several different ratings with varying rules and requirements.

The full list of assurance requirements for the Evaluation Assurance Levels is provided below:

* **EAL 1: The product is functionally tested; this is sought when some assurance in accurate operation is necessary, but the threats to security are not seen as serious.**
* **EAL 2: Structurally tested; this is sought when developers or users need a low to moderate level of independently guaranteed security.**
* **EAL 3: Methodically tested and checked; this is sought when there is a need for a moderate level of independently ensured security.**
* **EAL 4: Methodically designed, tested, and reviewed; this is sought when developers or users require a moderate to high level of independently ensured security.**
* **EAL 5: Semiformally designed and tested; this is sought when the requirement is for a high level of independently ensured security.**
* **EAL 6: Semiformally verified, designed, and tested; this is sought when developing specialized TOEs for high-risk situations.**
* **EAL 7: Formally verified, designed, and tested; this is sought when developing a security TOE for application in extremely high-risk situations.**

EALs are frequently misunderstood to provide a simple means to compare security products with similar levels. In fact, products may be very different even if they are assigned the same EAL level, since functionality may have little in common.

**SEI-CMM – Security Engineering Institute – Capability Maturity Model**

**Software Framework**

**The Capability Maturity Model (CMM)** is a service mark owned by Carnegie Mellon University (CMU) and refers to a development model elicited from actual data. The data was collected from organizations that contracted with the U.S. Department of Defense, who funded the research, and became the foundation from which CMU created the Software Engineering Institute (SEI). Like any model, it is an abstraction of an existing system.

**The Capability Maturity Model (CMM)** is a methodology used to develop and refine an organization's software development process. The model describes a five-level evolutionary path of increasingly organized and systematically more mature processes. CMM was developed and is promoted by the Software Engineering Institute (SEI), a research and development center sponsored by the U.S. Department of Defense (DoD). SEI was founded in 1984 to address software engineering issues and, in a broad sense, to advance software engineering methodologies. More specifically, SEI was established to optimize the process of developing, acquiring, and maintaining heavily software-reliant systems for the DoD. Because the processes involved are equally applicable to the software industry as a whole, SEI advocates industry-wide adoption of the CMM.

The CMM is similar to ISO 9001, one of the ISO 9000 series of standards specified by the International Organization for Standardization (ISO). The ISO 9000 standards specify an effective quality system for manufacturing and service industries; ISO 9001 deals specifically with software development and maintenance. The main difference between the two systems lies in their respective purposes: ISO 9001 specifies a minimal acceptable quality level for software processes, while the CMM establishes a framework for continuous process improvement and is more explicit than the ISO standard in defining the means to be employed to that end.

**CMM's Five Maturity Levels of Software Processes:**

**At the initial level**, processes are disorganized, even chaotic. Success is likely to depend on individual efforts, and is not considered to be repeatable, because processes would not be sufficiently defined and documented to allow them to be replicated.

**At the repeatable level**, basic project management techniques are established, and successes could be repeated, because the requisite processes would have been made established, defined, and documented.

**At the defined level,** an organization has developed its own standard software process through greater attention to documentation, standardization, and integration.

**At the managed level**, an organization monitors and controls its own processes through data collection and analysis.

**At the optimizing level,** processes are constantly being improved through monitoring feedback from current processes and introducing innovative processes to better serve the organization's particular needs.

**4.A.3 Supervisory Control and Data Acquisition (SCADA) (e.g., process automation networks, work interdependencies, monitoring requirements)**

**4.B Perform requirements analysis**

**4.B.1 Business and functional requirements(e.g. locations, jurisdictions, business sectors, cost, stakeholder preferences, quality attributes, capacity, manageability)**

**4.B.2 Treat modeling**

**4.B.3 Evaluate use cases (e.g. business rules and control objectives, misuse, abuse)**

**4.B.4 Gap analysis**

**4.B.5 Assess risk**

**4.B.6 Maturity models**

**4.C Design security architecture**

**4.C.1 Apply existing information security standards and guidelines (e.g. ISO/IEC, PCI, NIST)**

**4.C.2 Systems development Life Cycle (SDLC) (e.g. requirements traceability matrix, security architecture documentation, secure coding)**

**4.C.3 Application Security (e.g., Commercial Off-the-Shelf (COTS) integration)**

**4.D Verify and validate design**

**4.D.1 Validate threat model (e.g. access control attacks, cryptanalytic attacks, network attacks)**

**4.D.2 Evaluate controls against threats and vulnerabilities**

**4.D.3 Remediate gaps**

**4.D.4 Independent verification and validation**

**Accreditation** is the authorization by management to implement software or systems in a production environment. This authorization may be either provisional or full. Accreditation is the official management decision to operate a system. Accreditation is the formal declaration by a senior agency official (Designated Accrediting Authority (DAA) or Principal Accrediting Authority (PAA)) that an information system is approved to operate at an acceptable level of risk, based on the implementation of an approved set of technical, managerial, and procedural security controls (safeguards).

**Certification** is the process of evaluating the security stance of the software or system against a selected set of standards or policies. Certification is the technical evaluation of a product. This may precede accreditation but is not a required precursor. Certification is the process of performing a comprehensive analysis of the security features and safeguards of a system to establish the extent to which the security requirements are satisfied. Shon Harris states in her book that Certification is the comprehensive technical evaluation of the security components and their compliance for the purpose of accreditation.

**Evaluation** as a general term is described as the process of independently assessing a system against a standard of comparison, such as evaluation criteria. Evaluation criterias are defined as a benchmark, standard, or yardstick against which accomplishment, conformance, performance, and suitability of an individual, hardware, software, product, or plan, as well as of risk-reward ratio is measured.

**Acceptance testing** refers to user testing of a system before accepting delivery.

**--------------------**

**OTHERS**

AIO defines **Compartmented Mode Workstation (CMW)** as a workstation that contains the necessary controls to be able to operate as a trusted computer. The system is trusted to keep data from different classification levels and categories in separate compartments and properly protected.

**Information Labels** are similar to Sensitivity Labels, but in addition to the classification and the category set of the Sensitivity Labels, they also have the necessary controls to be able to operate as a trusted computer.

The AIO defines **Object Reuse** as reassigning to a subject media that previously contained information. Object reuse is a security concern because if insufficient measures were taken to erase the information on the media, the information may be disclosed to unauthorized personnel. This is a big issue when you think of shared resources such as RAM where a portion might be assigned to an application ten reassigned to another application a bit later. The memory manager must do a full refresh of the memory area before it is reassigned to another application.

Allowing objects to be used sequentially by multiple users without a refresh of the objects can lead to disclosure of residual data. It is important that steps be taken to eliminate the chance for the **disclosure of residual data.**

**Ring Architecture**

In computer science, hierarchical protection domains, often called protection rings, are a mechanism to protect data and functionality from faults (fault tolerance) and malicious behavior (computer security). This approach is diametrically opposite to that of capability-based security.

Computer operating systems provide different levels of access to resources. A protection ring is one of two or more hierarchical levels or layers of privilege within the architecture of a computer system. This is generally hardware-enforced by some CPU architectures that provide different CPU modes at the hardware or microcode level. Rings are arranged in a hierarchy from most privileged (most trusted, usually numbered zero) to least privileged (least trusted, usually with the highest ring number). On most operating systems, Ring 0 is the level with the most privileges and interacts most directly with the physical hardware such as the CPU and memory.

Special gates between rings are provided to allow an outer ring to access an inner ring's resources in a predefined manner, as opposed to allowing arbitrary usage. Correctly gating access between rings can improve security by preventing programs from one ring or privilege level from misusing resources intended for programs in another. For example, spyware running as a user program in Ring 3 should be prevented from turning on a web camera without informing the user, since hardware access should be a Ring 1 function reserved for device drivers. Programs such as web browsers running in higher numbered rings must request access to the network, a resource restricted to a lower numbered ring.

**Trusted Computing Base (TCB**) is responsible for providing security. Limited Privilege is assured by isolating the resources that are protected . Processes running in one domain often invoke processes in other domains to obtain more sensitive data or services. The TCB ensures that the processes provide security by granting minimal privilege, not more than needed.

The Trusted Computing Base (TCB) is the totality of protection mechanisms within a computer system — including hardware, firmware, software, processes, and some inter-process communications — that, combined, are responsible for enforcing a security policy.

TCB is:

* It is defined in the Orange Book.
* It includes hardware, firmware and software.
* A higher TCB rating will require that details of their testing procedures and documentation be reviewed with more granularity.

**DIACAP**

DITSCAP has been replaced by DIACAP (DoD Information Assurance Certification and Accreditation Process) effective Nov 2007 for C&A within the Department of Defense.

The DoD Information Assurance Certification and Accreditation Process (DIACAP) is the United States Department of Defense (DoD) process to ensure that risk management is applied on information systems (IS). DIACAP defines a DoD-wide formal and standard set of activities, general tasks and a management structure process for the certification and accreditation (C&A) of a DoD IS that will maintain the information assurance (IA) posture throughout the system's life cycle.

An interim version of the DIACAP was signed July 6, 2006 and superseded DITSCAP. The final version is titled Department of Defense Instruction 8510.01 and was signed on November 28, 2007. It supersedes the Interim DIACAP Guidance.

**NIACAP**

National Information Assurance Certification and Accreditation Process (NIACAP), establishes the minimum national standards for certifying and accrediting national security systems. This process provides a standard set of activities, general tasks, and a management structure to certify and accredit systems that will maintain the Information Assurance (IA) and security posture of a system or site.

**HIPAA**

The HIPAA legislation had four primary objectives:

(1) Assure health insurance portability by eliminating job-lock due to pre-existing medical conditions,

(2) Reduce healthcare fraud and abuse,

(3) Enforce standards for health information and

(4) Guarantee security and privacy of health information.

**NIST**

The correct answer is ' The National Institute of Standards and Technology (NIST) ' as FIPS publications are issued by NIST after approval by the Secretary of Commerce pursuant to Section 5131 of the Information Technology Reform Act of 1996, Public Law 104-106, and the FISMA Act of 2002.

As per FDA **data should be attributable**, original, accurate, contemporaneous and legible. In an automated system attributability could be achieved by a computer system designed to identify individuals responsible for any input.

**The reference monitor** refers to abstract machine that mediates all access to objects by subjects.

This question is asking for the concept that governs access by subjects to objects, thus the reference monitor is the best answer. While the security kernel is similar in nature, it is what actually enforces the concepts outlined in the reference monitor.

In operating systems architecture a reference monitor concept defines a set of design requirements on a reference validation mechanism, which enforces an access control policy over subjects' (e.g., processes and users) ability to perform operations (e.g., read and write) on objects (e.g., files and sockets) on a system. The properties of a reference monitor are:

The reference validation mechanism must always be invoked (complete mediation). Without this property, it is possible for an attacker to bypass the mechanism and violate the security policy.The reference validation mechanism must be tamperproof (tamperproof). Without this property, an attacker can undermine the mechanism itself so that the security policy is not correctly enforced.The reference validation mechanism must be small enough to be subject to analysis and tests, the completeness of which can be assured (verifiable). Without this property, the mechanism might be flawed in such a way that the policy is not enforced.

**Reference Monitor is used to compare the security labels of a subject and an object.**

**The reference monitor must meet three conditions:**

**(1) it must be tamperproof (isolation)**

**(2) it must be invoked on every access to every object (completeness) and**

**(3) it must be small enough for thorough validation of its operation through analysis and tests, in order to verify completeness (verifiability).**

**Process isolation** is where each process has its own distinct address space for its application code and data. In this way, it is possible to prevent each process from accessing another process' data. This prevents data leakage, or modification to the data while it is in memory. Memory segmentation is a virtual memory management mechanism. The reference monitor is an abstract machine that mediates all accesses to objects by subjects. Data hiding, also known as information hiding, is a mechanism that makes information available at one processing level is not available at another level.

**In dedicated security mode**, all users have a clearence and formal need to know for all information processed within the system. If for example, the system stores and processes top secret information on the "Nova Widgets" program, then all users of the system must have both a top secret clearence and a formal need to know for the "Nova Widgets" program.

**Trusted System** does:

* **Enforcement of a security policy.**
* **Sufficiency and effectiveness of mechanisms to be able to enforce a security policy.**
* **Independently-verifiable evidence that the security policy-enforcing mechanisms are sufficient and effective.**

**Goals of Integrity:**

* **Prevent unauthorized users from making modifications.**
* **Maintain internal and external consistency.**
* **Prevent authorized users from making improper modifications.**

**CISC:**

**Complex Instruction Set Computer (CISC)** uses instructions that perform many operations per instruction. It was based on the fact that in earlier technologies, the instruction fetch was the longest part of the cycle. Therefore, by packing more operations into an instruction, the number of fetches could be reduced. Pipelining involves overlapping the steps of different instructions to increase the performance in a computer. Reduced Instruction Set Computers (RISC) involve simpler instructions that require fewer clock cycles to execute. Scalar processors are processors that execute one instruction at a time.

**Pipelining** is a natural concept in everyday life, e.g. on an assembly line. Consider the assembly of a car: assume that certain steps in the assembly line are to install the engine, install the hood, and install the wheels (in that order, with arbitrary interstitial steps). A car on the assembly line can have only one of the three steps done at once. After the car has its engine installed, it moves on to having its hood installed, leaving the engine installation facilities available for the next car. The first car then moves on to wheel installation, the second car to hood installation, and a third car begins to have its engine installed. If engine installation takes 20 minutes, hood installation takes 5 minutes, and wheel installation takes 10 minutes, then finishing all three cars when only one car can be assembled at once would take 105 minutes. On the other hand, using the assembly line, the total time to complete all three is 75 minutes. At this point, additional cars will come off the assembly line at 20 minute increments.

In computing, a pipeline is a set of data processing elements connected in series, so that the output of one element is the input of the next one. The elements of a pipeline are often executed in parallel or in time-sliced fashion; in that case, some amount of buffer storage is often inserted between elements. Pipelining is used in processors to allow overlapping execution of multiple instructions within the same circuitry. The circuitry is usually divided into stages, including instruction decoding, arithmetic, and register fetching stages, wherein each stage processes one instruction at a time.

**Very-Long Instruction-Word Processor (VLIW) -** a computer processing architecture in which a language compiler or pre-processor breaks program instructions down into basic operations that can be performed by the processor at the same time.

**A fault-tolerant system** is capable of detecting that a fault has occurred and has the ability to correct the fault or operate around it. In a fail-safe system, program execution is terminated, and the system is protected from being compromised when a hardware or software failure occurs and is detected. In a fail-soft system, when a hardware or software failure occurs and is detected, selected, non-critical processing is terminated. The term failover refers to switching to a duplicate "hot" backup component in real-time when a hardware or software failure occurs, enabling processing to continue.

**Fail over -** When one system/application fails, operations will automatically switch to the backup system.

**Fail safe -** Pertaining to the automatic protection of programs and/or processing systems to maintain safety when a hardware or software failure is detected in a system.

**Fail secure -** The system preserves a secure state during and after identified failures occur.

**Fail soft -** Pertaining to the selective termination of affected non-essential processing when a hardware or software failure is detected in a system

**The Graham–Denning** is primarily concerned with how subjects and objects are created, how subjects are assigned rights or privileges, and how ownership of objects is managed. In other words, it is primarily concerned with how a model system controls subjects and objects at a very basic level where other models simply assumed such control.

**The Graham–Denning** access control model has three parts: a set of objects, a set of subjects, and a set of rights.

"The Graham—Denning access control model defines eight primitive protection rights:

1. Create object: the ability to create a new object

2. Create subject: the ability to create a new subject

3. Delete object: the ability to delete an existing object

4. Delete subject: the ability to delete an existing subject

5. Read access right: the ability to view current access privileges

6. Grant access right: the ability to grant access privileges

7. Delete access right: the ability to remove access privileges

8. Transfer access right: the ability to transfer access privileges from one subject or object to another subject or object"

**"system high" security policy -** system where all users are cleared to view the most highly classified data on the system.

**Configuration Management as:** The use of procedures appropriate for controlling changes to a system’s hardware, software, or firmware structure to ensure that such changes will not lead to a weakness or fault in the system.

**TCSEC** focused on confidentiality while **ITSEC** added integrity and availability as security goals.

**Information Technology Security Evaluation Criteria (ITSEC).**Following the release of the Orange Book, several European countries issued their own criteria. The European Information Technology Security Evaluation Criteria (ITSEC) was first drafted in 1990 and became endorsed by the Council of the European Union in 1995. It was the result of Dutch, English, French, and German activities designed to define and harmonize European national security evaluation criteria. Although ITSEC includes concepts from the TCSEC, the Orange Book was thought to be too rigid. Thus, ITSEC attempted to provide a framework for security evaluation that would be more flexible. One major difference between the two is ITSEC’s inclusion of integrity and availability as security goals, along with confidentiality.

**Data mining** is used to reveal hidden relationships, patterns and trends by running queries on large data stores.

**How to protect objects and the data within the object?**

**Layering** assigns specific functions to each layer and communication between layers is only possible through well-defined interfaces. This helps preclude tampering in violation of security policy.

**Abstraction** "hides" the particulars of how an object functions or stores information and requires the object to be manipulated through well-defined interfaces that can be designed to enforce security policy.

**Data hiding** conceals the details of information storage and manipulation within an object by only exposing well defined interfaces tot he information rather than the information itslef. For example, the details of how passwords are stored could be hidden inside a password object with exposed interfaces such as check\_password, set\_password, etc. When a password needs to be verified, the test password is passed to the check\_password method and a boolean (true/false) result is returned to indicate if the password is correct without revealing any details of how/where the real passwords are stored.

**The noninterference model** is the correct answer. The goal of a noninterference model is to strictly separate differing security levels to assure that higher-level actions do not determine what lower-level users can see. This is in contrast to other security models that control information flows between differing levels of users, By maintaining strict separation of security levels, a noninterference model minimizes leakages that might happen through a covert channel.

**An over channel** is a path within a computer system or network that is designed for the authorized transfer of data. The opposite would be a covert channel which is an unauthorized path.

**A covert channel** is a way for an entity to receive information in an unauthorized manner. It is an information flow that is not controlled by a security mechanism. This type of information path was not developed for communication; thus, the system does not properly protect this path, because the developers never envisioned information being passed in this way. Receiving information in this manner clearly violates the system’s security policy.

**A security domain** is a domain of trust that shares a single security policy and single management.

The term security domain just builds upon the definition of domain by adding the fact that resources within this logical structure (domain) are working under the same security policy and managed by the same group.

So, a network administrator may put all of the accounting personnel, computers, and network resources in Domain 1 and all of the management personnel, computers, and network resources in Domain 2. These items fall into these individual containers because they not only carry out similar types of business functions, but also, and more importantly, have the same type of trust level. It is this common trust level that allows entities to be managed by one single security policy.

The different domains are separated by logical boundaries, such as firewalls with ACLs, directory services making access decisions, and objects that have their own ACLs indicating which individuals and groups can carry out operations on them.

All of these security mechanisms are examples of components that enforce the security policy for each domain. Domains can be architected in a hierarchical manner that dictates the relationship between the different domains and the ways in which subjects within the different domains can communicate. Subjects can access resources in domains of equal or lower trust levels.

**Security Kernel**

* **The security kernel is made up of mechanisms that fall under the TCB and implements and enforces the reference monitor concept.**
* **The security kernel must provide isolation for the processes carrying out the reference monitor concept and they must be tamperproof.**
* **The security kernel must be small enough to be able to be tested and verified in a complete and comprehensive manner.**

**Indirect addressing** is when the address location that is specified in the program instruction contains the address of the final desired location. Direct addressing is when a portion of primary memory is accessed by specifying the actual address of the memory location. Indexed addressing is when the contents of the address defined in the program's instruction is added to that of an index register. Program addressing is not a defined memory addressing mode.

**The security perimeter** is the imaginary line that separates the trusted components of the kernel and the Trusted Computing Base (TCB) from those elements that are not trusted. The reference monitor is an abstract machine that mediates all accesses to objects by subjects. The security kernel can be software, firmware or hardware components in a trusted system and is the actual instantiation of the reference monitor. The reference perimeter is not defined and is a distracter.

**Simple Security Property and Polyinstantiation**

**The Simple Security Property** states that a subject at a given clearance may not read an object at a higher classification, so unclassified APFEL could not read FIGCO's top secret cargo information.

**Polyinstantiation** permits a database to have two records that are identical except for their classifications (i.e., the primary key includes the classification). Thus, APFEL's new unclassified record did not collide with the real, top secret record, so APFEL was not able to learn about FIGs pineapples.

**The Brewer and Nash model** was constructed to provide information security access controls that can change dynamically. This security model, also known as the Chinese wall model, was designed to provide controls that mitigate conflict of interest in commercial organizations, and is built upon an information flow model.

In the Brewer and Nash Model no information can flow between the subjects and objects in a way that would create a conflict of interest.

**Middleware** is a connectivity software that enables multiple processes running on one or more machines to interact.

**Pervasive Computing and Mobile Computing Devices**

* These devices share common security concerns with other resource-constrained devices.
* In many cases, security services have been sacrificed to provide richer user interaction when processing power is very limited.
* Their mobility has made them a prime vector for data loss since they can be used to transmit and store information in ways that may be difficult to control.

**Compartmented security mode** allows a system to have differing levels of information and all users have clearence to access all the information (which means they must be cleared to access the highest level of information stored and processed on the sytem) but not all users have the need to know.

**Multilevel security mode.** In multi-level mode, the system stores and processes information of differing classifications but users are only required to have clearence and need to know for the information they need to access. For example, multilevel security would allow a system that processes confidential and secret information to be accessed by users with clearences of confidential and secret but the system would impose controls to assure that each user only accesses information for which they have both clearence and formal need to know.

**System-high security mode.** System-high security mode requires all users to have a clearence for the level of information stored and processed by the system but all users have a need to know for only some of the data stored and processed by the system. For example, though all information on "Nova Widgets" is top secret and all users of the system must therefore have a top secret clearence, some users may only have a need to know for "Nova Widgets Ordnance" information while other users may only have a need to know for the "Nova Widgets Guidance and Propulsion" information.

**Dedicated security mode.** In dedicated security mode, all users have a clearence and formal need to know for all information processed within the system. If for example, the system stores and processes top secret information on the "Nova Widgets" program, then all users of the system must have both a top secret clearence and a formal need to know for the "Nova Widgets" program.