11TH INTERNATIONAL COMMAND AND CONTROL RESEARCH AND TECHNOLOGY SYMPOSIUM -- COALITION COMMAND AND CONTROL IN THE NETWORKED ERA

Modeling Security Architectures for the Enterprise

STUDENT PAPER

MAJOR GEORGE C. DALTON II*
DR. JOHN COLOMBI
DR. ROBERT MILLS

Air Force Institute of Technology
Department of Electrical and Computer Engineering
2950 Hobson Way
Wright-Patterson Air Force Base, Ohio
937-255-3636 x7556/fax 937-255- 656-7342
George.Dalton@afit.edu
John.Colombi@afit.edu
Robert.Mills@afit.edu

DISCLAIMER
The views expressed in this academic research paper are those of the author(s) and do not reflect the official policy or position of the US government or the Department of Defense. In accordance with Air Force Instruction 51-303, it is not copyrighted, but is the property of the United States government.

Modeling Security Architectures for the Enterprise

George Dalton George.Dalton@afit.edu

Dr. John Colombi John.Colombi@afit.edu

Dr. Bob Mills Robert.Mills@afit.edu

Abstract

Security is often treated, whether intentionally or otherwise, as something which can easily be added after a system is built. This has proven in many cases to cause substantial embarrassment and cost. Security must be designed into a system or enterprise, and integrated across all subsystems from the beginning. However, the tools and methodologies to perform security-related design are not well developed, defined, standardized, nor integrated. There are many possible ways that security can be modeled. Part of the problem is that security is sometimes considered a nonfunctional or performance system requirement, sometimes a functional system requirement, and sometimes simply an operational mission requirement. This paper explores ways to model each of these using current methodologies, and suggests a unifying theme for modeling security as a whole. Descriptions of how security can be modeled in DoDAF using structured and object oriented techniques are presented, how it is modeling in other frameworks and how the use of Colored Petri-Nets can be a potential candidate for security architectures.

Introduction

In a perfect world where no malicious or accidental acts could affect operations, security would be an unnecessary expense. However, this not the case as evidenced by continued reports of computer attacks including the recent compromise and theft of 33,000 personnel records from the U.S. Air Force Assignment Management System (AMS). So we must expend some resources to design appropriate levels of security measures. For security within an information age, it is no surprise we are primarily concerned with the protection of our information. However, the usefulness of information is increased when it is shared, often within a community of interest, and amalgamated with other sources. This sharing of information spawns many security related questions. Where are all my security components? Where are the secure or trusted boundaries? How do I control access to particular assets? How do I know my controls are working? What is my overall security architecture? Can I incrementally fix security later?

While adding security after the fact can be done and occurs regularly as the IT community patches vulnerabilities, it becomes a much more difficult and expensive task. This fact is true of most designs, especially software, which get incrementally modified or "fixed". It is well known that these fixes deteriorate the software product, and eventually the integrity of the software architecture is lost as shown in Figure 1.

With the exponential growth of the Internet in the past decade, including vulnerabilities and attacks, and its amazing transformation of business, banking, government and warfighting, there has been ever increasing concern over Internet Security and its design. It has become apparent that security must be an integral part of any system design that is intended to function properly and reliably in today's networked environment.

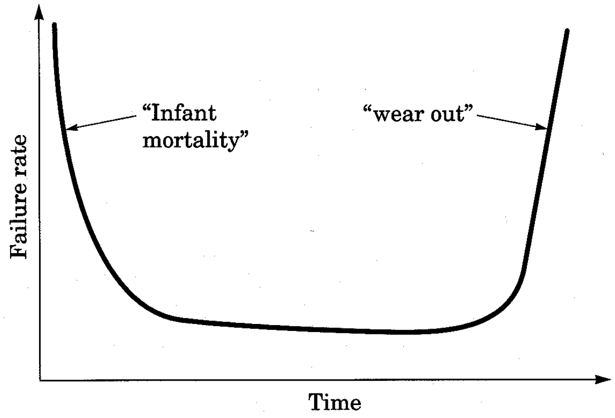
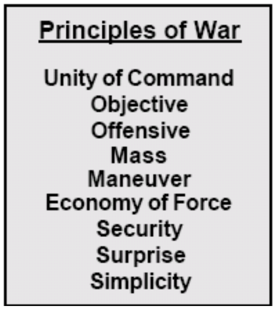


Figure 1 - Failure of software architectures over time

With the exponential growth of the Internet in the past decade, including vulnerabilities and attacks, and its amazing transformation of business, banking, government and warfighting, there has been ever increasing concern over Internet Security and its design. It has become apparent that security must be an integral part of any system design that is intended to function properly and reliably in today's networked environment.

Security and Information Assurance

What is Security? According to Webster, Security includes "measures taken to guard against espionage or sabotage, crime, attack, or escape" . Air Force doctrine further defines the purpose of security as to "never permit the enemy to acquire unexpected advantage"(USAF, 2003). It calls for protection of friendly forces and operations, and reducing vulnerability to attack. It is so important for military forces that it is considered one of the nine basic principles of war (Figure 2).



Principles of War

Unity of Command
Objective
Offensive
Mass
Maneuver
Economy of Force
Security
Surprise
Simplicity

Figure 2 - Principles of War (USAF, 2003)

In today's high technology environment with the rapid transition to NetCentric warfare it is essential to understand that security encompasses both physical security and security of the information medium. This "security of the information medium" is the objective of Information Assurance.

What is Information Assurance (IA)? Air Force Information Operations doctrine states that IA is vital to operational readiness, and defines it as the "continuous integration of trained personnel, operational and technical capabilities and necessary policies and procedures to guarantee ***availability, integrity, authenticity, confidentiality,*** and ***non-repudiation*** of information services, while providing the means to efficiently reconstitute these vital services following disruptions of any kind, whether from an attack, natural disaster, equipment failure, or operator error" (USAF, 2005). This is an expansion of the industry standard CIA – Confidentiality, Integrity, and Availability – model. Confidentiality is the property where the privacy of the message is preserved along the entire path from sender to receiver as shown in Figure 3, and can be further restricted to only those authorized access throughout the lifecycle of the data being protected. The lifecycle of data begins with its creation and ends with its destruction. Integrity is the property that the message has not changed throughout the transmission from sender to receiver (Figure 4), and further, that the data is never maliciously or inadvertently modified during its lifecycle. Availability is the property where the message or data is ready when needed. Denial of service attacks are intended to reduce or eliminate availability. Authenticity is the property of identifying a user or process in such a way that the recipient of a message (Figure 5) or potential partner in an operation is certain that the user or process is who it says it is. In the CIA model, this property would be covered under a combination of Integrity and Confidentiality. Similarly, non-repudiation, not only authenticates the user, but ensures that the user or process cannot deny at a later time that it sent the message or initiated a process.

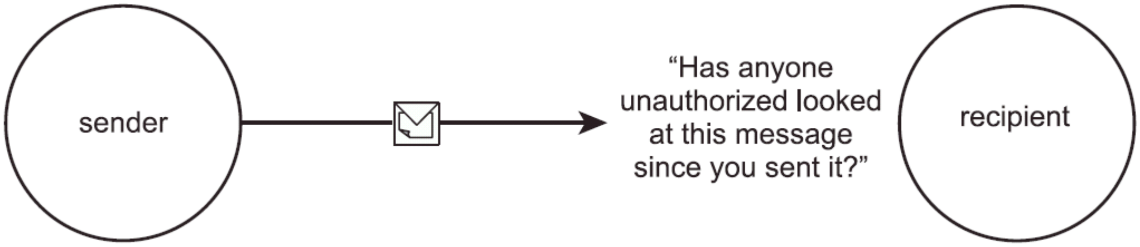

sender
"Has anyone unauthorized looked at this message since you sent it?"
recipient

Figure 3 - Confidentiality: Privacy of Message preserved throughout the Message Path (Erl, 2005)

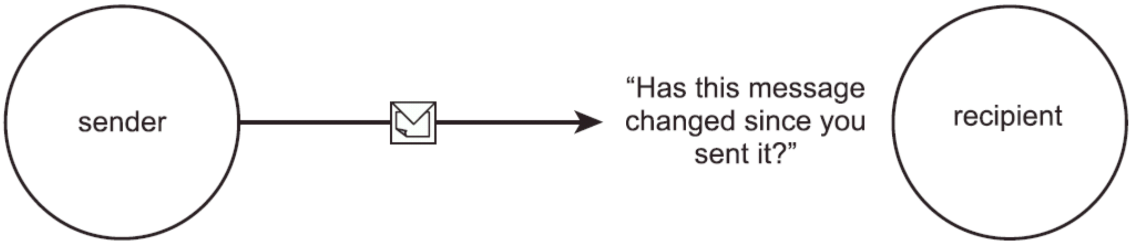


Figure 4 - Integrity: Message not changed during Transmission (Erl, 2005)

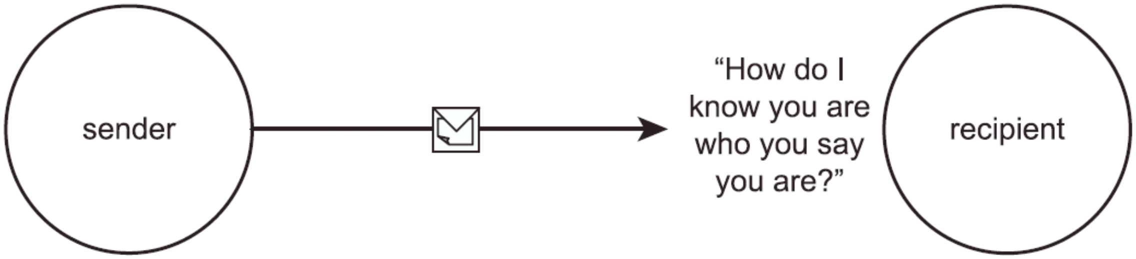


Figure 5 – Authentication: Proving Identity (Erl, 2005)

Another hybrid concept that is covered using the previous definitions is Authorization (Figure 6). This says that not only is the user or process who it says it is, but that it has permission to perform the operation (read, write, execute, etc.) on the object it is attempting to access.

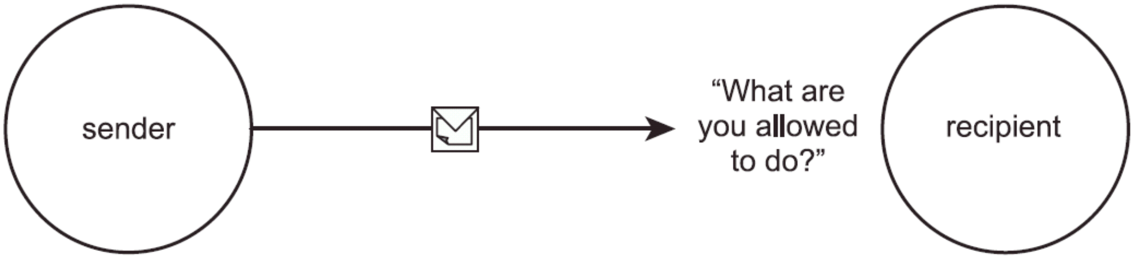


Figure 6 - Authorization: What extent does Authentication Apply and what can this Sender do? (Erl, 2005)

Cost of Security

The Department of Defense (DoD) asked the Institute for Defense Analyses (IDA) to help estimate the costs for physical security, information and information system security, personnel security, counterintelligence and law enforcement, and selected cross-disciplinary activities (IDA, 2004). In their report, IDA states "Historically, these costs have been difficult to pin down, because they result from decisions - often not well documented - made by unit commanders throughout the Department. As a result, funding for security activities is spread across many program elements in the DoD budget, rather than being concentrated in a handful of focused efforts." IDA estimated the DoD cost for asset protection to be about $11.6 billion in 2004, with about $4.8 billion of that earmarked for information protection alone. This is a huge expense for government and industry alike, and the cost of security failure is extremely high as well. For example, with only 639 respondents to the CSI/FBI Computer Crime and Security Survey, the reported loss for 2004 alone was over $130 million as shown in Figure 7 (Lucyshyn and Richardson., 2005). Many commercial firms are reluctant to report loss due to fears that it would affect their corporate image hurting stock prices, or otherwise be used to the advantage of their competitors.

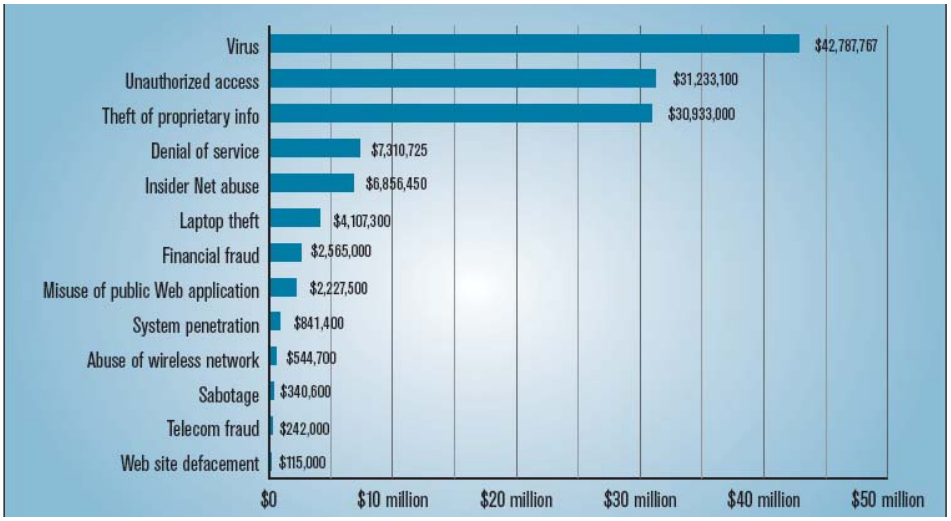


Figure 7 – Reported Costs of Computer Attacks by Type in 2005 (Lucyshyn and Richardson., 2005). Total losses are estimated to be over $130 million.

Net-centricity

The cost of failure of security is not merely monetary. The DoD is becoming more and more dependent on networks to control every aspect of their fighting forces, and so the security of the US is directly being tied to the security of our key networks. Recent news has spotlighted the large number computer attacks coming from China. Whether the attacks are an organized effort or individual efforts, this type of threat will continue until it is no longer successful or profitable. The drive toward a network centric military infrastructure is pressing the envelope of communications and information assurance. Defense Information Systems Agency's roadmap of key net-centric initiatives includes the following:

- Reducing Bandwidth Constraints
    - Joint Tactical Radio System: IP-based, self-managed, Beyond Line-of-sight (BLOS), mobile data and voice communication service
    - Global Information Grid Bandwidth (GIG) Expansion: ubiquitous, secure, robust, optional IP foundation (backbone) network
    - Transformational Communications Architecture (SATCOM): Supports mobile/ tactical users and global intelligence through optical cross-links and Extremely High-Frequency (EHF) IP links
- Deploy Trusted Services
    - Network-Centric Enterprise Services: Core information and data services include messaging: collaborations, mediation, storage, discovery, security, access, network management and specific Community of Interest (COI) applications
    - Information Assurance: Enables trusted computer, networking, and data services to all GIG users
- Improve Situational Awareness
    - Horizontal Fusion: Prototyping the means / tools to enable the smart pull and fusion of data by GIG users and application

The Deploy Trusted Services initiative, composed of Enterprise Services and Information Assurance, is a key enabler of net-centricity. These Network Core Enterprise Services (NCES) support both long-standing COIs and temporary (transient) COI representing missions or joint task forces as shown in Figure 8. Without a secure infrastructure, net centric warfare will not be possible. Not only are the monetary and operation requirements for providing a secure infrastructure, but there are also a number of laws (i.e.: Sarbanes-Oxley Act), executive directives and policies that mandate requirements for protecting government and certain commercial systems.

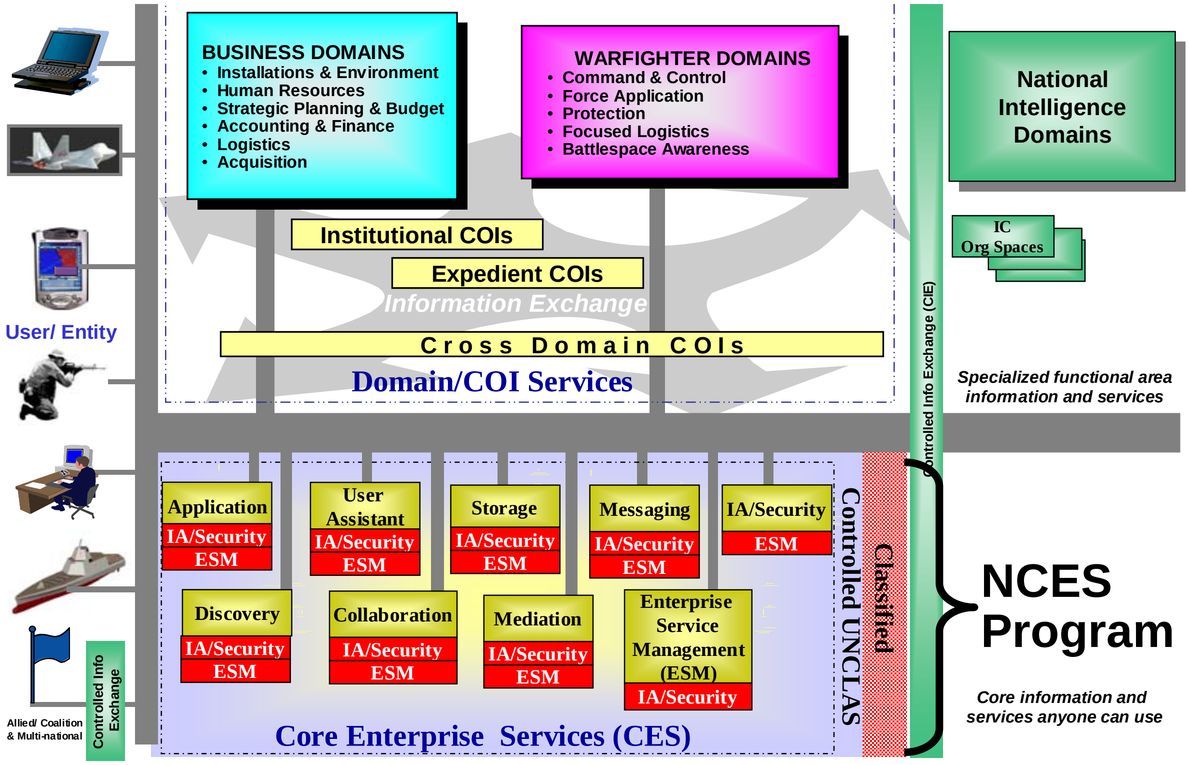


Figure 8 - Net-Centric Enterprise Services (Raduege, 2004)

Requirement Types

Throughout the literature, there appears to be three primary ways to capture security requirements: security as a nonfunctional or performance requirement. a functional requirement, or an operational requirement.

Security as a Nonfunctional/ Performance requirement

Sometimes security requirements are considered nonfunctional. This is the case in the requirement “Classified document icons shall be red.” These requirements tend to use “Security” as a descriptive adjective. They are not operational or performance based, but are non-the-less important in the definition of a system. Sometimes the word “Security” itself is used to specify a nonfunctional requirement as in “The secure data shall be stored for 90 days.”

Safety is a lot like Security in many ways. You can never be completely safe for all circumstances. For instance, if you triple lock your house and install a home security system with telephone connection to a central office, but then a hurricane floods your neighborhood. It could be said that your attempt to be safe failed. However, your safety feature, in this case the home security systems was only intended to keep criminals from entering your house, not flood waters.

Some security requirements are performance based. For Example, “100% of traffic transmitted shall be encrypted using AES-CCMP.” These type requirements tend to use “Security” as a measure or metric. Performance based requirements are many times dynamic and can be a challenge to model and verify.

Security as a Functional requirement

Within a functional decomposition, one may choose to add functional requirements to capture security behaviors. For example, “encrypt data”,” digitally sign an email message” or “authorize users” could be found in a node tree. These requirements tend to use security as a verb or verb phrase.

Security as a Operational Requirement

The military has unique missions related to network security that are not normally shared by individuals and corporations. The following definitions are from Air Force Doctrine on Information Operations (USAF, 2005):

Network warfare operations are the integrated planning, employment, and assessment of military capabilities to achieve desired effects across the interconnected analog and digital network portion of the battlespace. Network warfare operations are conducted in the information domain through the combination of hardware, software, data, and human interaction. Networks in this context are defined as any collection of systems transmitting information. …The operational activities of network warfare operations are network attack (NetA), network defense (NetD) and network warfare support (NS) (USAF, 2005).

Network attack (NetA) is employment of network-based capabilities to destroy, disrupt, corrupt, or usurp information resident in or transiting through networks. Networks include telephony and data services networks.

Network defense (NetD) is employment of network-based capabilities to defend friendly information resident in or transiting through networks against adversary efforts to destroy, disrupt, corrupt, or usurp it.

Network warfare support (NS) is the collection and production of network related data for immediate decisions involving NW Ops. NS is critical to NetA and NetD actions to find, fix, track, and assess both adversaries and friendly sources of access and vulnerability for the purpose of immediate defense, threat prediction and recognition, targeting, access and technique development, planning, and execution in NW Ops.

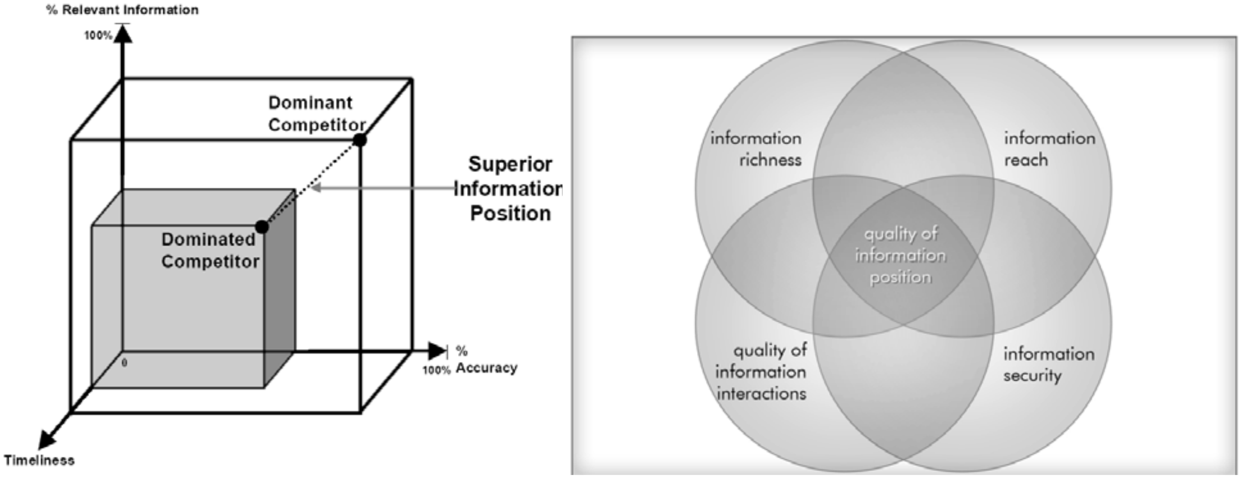


Figure 9 - Superior Information Position (Alberts et al., 2003, Alberts and Hayes, 2006)

The goal of Network Warfare Operations is to achieve Information Superiority, which is defined as "a state that is achieved when a competitive advantage is derived from the ability to exploit a superior information position (Figure 9)" (Alberts et al., 2003, Alberts and Hayes, 2006). The left graphic portrays a superior or dominant information position, regarding the timeliness, accuracy and relevance of enterprise information. Naturally, the ability to delay critical information, modify content or inundate a competitor with misinformation would weaker one's position. The right graphic shows that the quality of your information is based on it's richness, reach, interactions, and security. We are just beginning to realize the potential of net-centric operations. Figure 10 shows a network enabled sensor-to-shooter configuration implemented and used during Operation Iraqi Freedom. It is a far improvement from what was used during Dessert Storm, but is still a small improvement compared to what will be possible in a fully network centric environment. However, to achieve improvements we must continue to solve many security related problems of authentication, authorization, integrity, confidentiality, etc. In other words we need a security architecture based on these Information Assurance fundamentals.

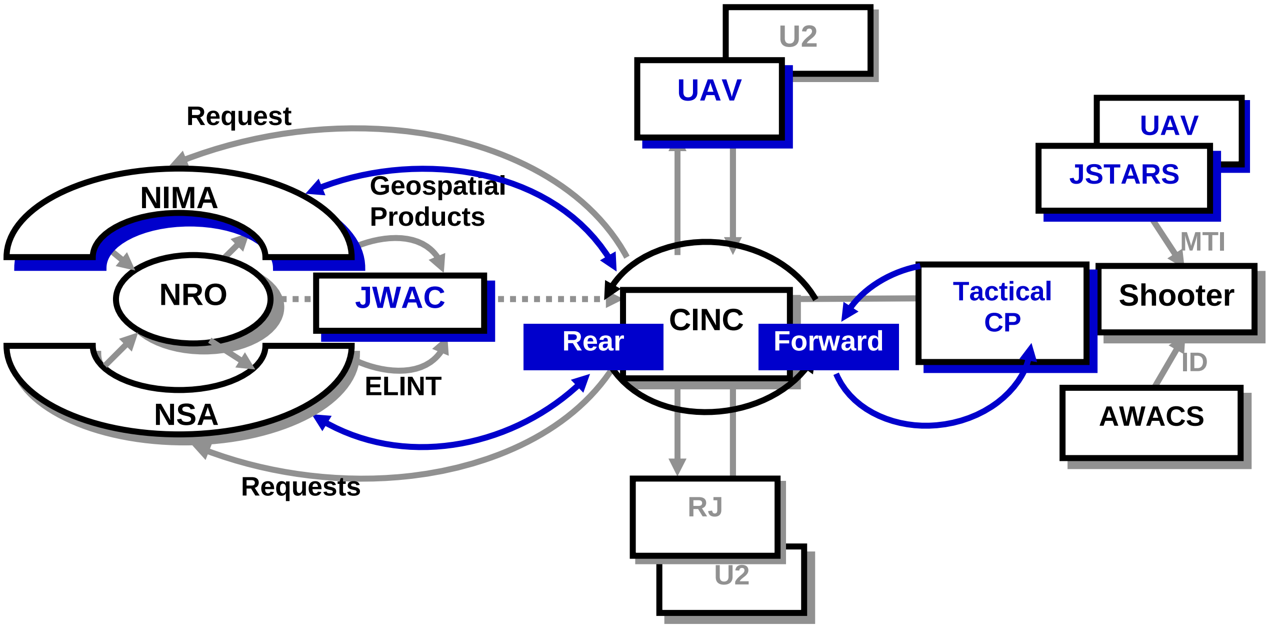


Figure 10 - Net-Centric Infancy (Taylor, 2004)

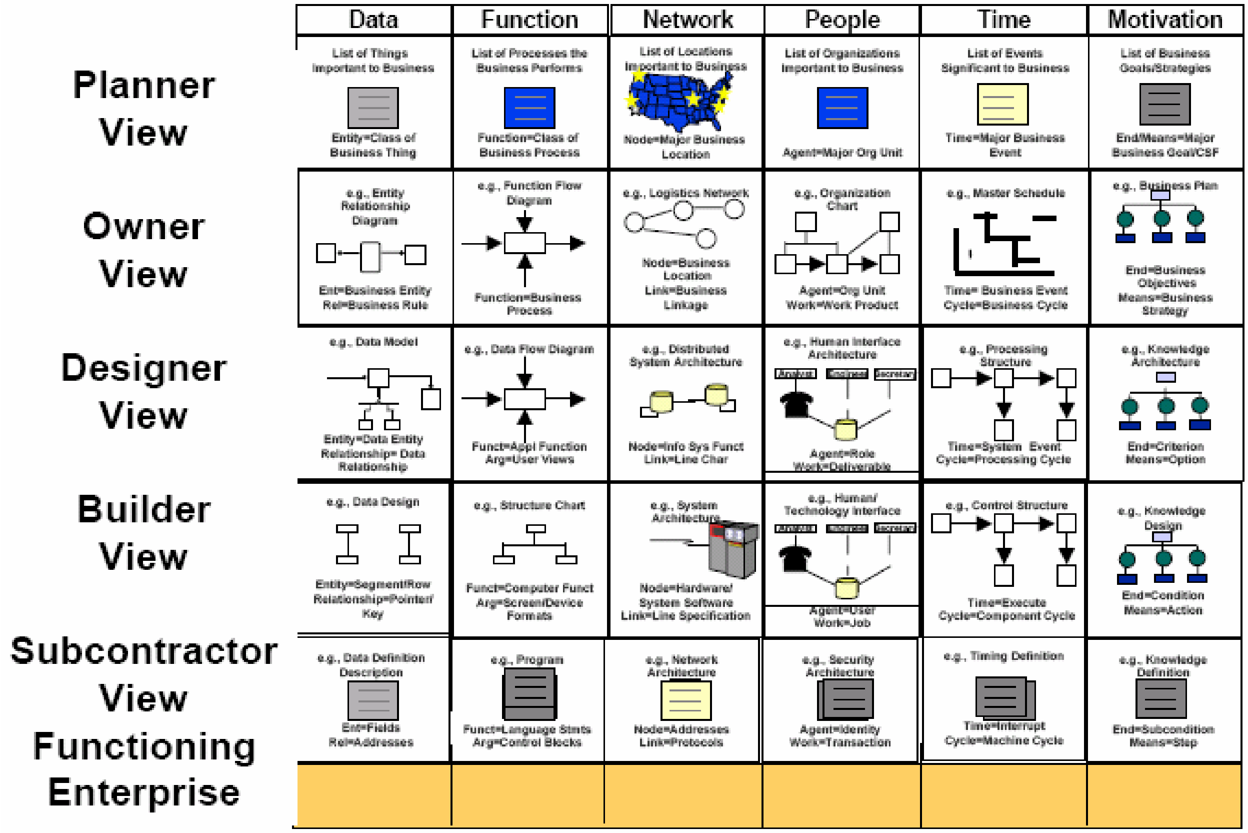
Security Frameworks, Architectures and Models

Architecting security solutions for information systems have seen a long historic and wide variety of modeling approaches, including Frameworks, Architectures, and Models. Architecture Frameworks define a conceptual model which provides the **rules, guidance, and product descriptions** for developing and presenting architecture that ensure **inter-relatable** and integrated descriptions across organizational boundaries. On the other hand, Architecture is defined as the structure of components, their relationships, and the principles and guidelines governing their design and evolution over time [DoDAF]. These "components" can be any aspect of the overall design which needs to be communicated or documented. Most of the work in security architectures has been resident within the taxonomy of a representational models and their accompanying analysis.

Frameworks

Zachman Framework

John Zachman, from IBM, developed the first information systems framework in 1987. Several other Frameworks have followed in the last two decades. The Zachman Framework is an Enterprise-level analytical tool for ensuring that important issues in the planning and development of information systems are not overlooked. The framework does this by enumerating both the players or roles and the interrogatives (what, how, where, why, etc). For enterprise information systems, these interrogatives represent information/data, networks, organizations/people, functions, temporal/dynamics and motivation as shown in Figure 11.



	Data	Function	Network	People	Time	Motivation
Planner View	List of Things Important to Business Entity=Class of Business Thing	List of Processes the Business Performs Function=Class of Business Process	List of Locations Important to Business Node=Major Business Location	List of Organizations Important to Business Agent=Major Org Unit	List of Events Significant to Business Time=Major Business Event	List of Business Goals/Strategies End/Means=Major Business Goal/CSF
Owner View	e.g., Entity Relationship Diagram Ent=Business Entity Rel=Business Rule	e.g., Function Flow Diagram Function=Business Process	e.g., Logistics Network Node=Business Location Link=Business Linkage	e.g., Organization Chart Agent=Org Unit Work=Work Product	e.g., Master Schedule Time= Business Event Cycle=Business Cycle	e.g., Business Plan End=Business Objectives Means=Business Strategy
Designer View	e.g., Data Model Entity=Data Entity Relationship= Data Relationship	e.g., Data Flow Diagram Funct=Appl Function Arg=User Views	e.g., Distributed System Architecture Node=Info Sys Funct Link=Line Char	e.g., Human Interface Architecture Agent=Role Work=Deliverable	e.g., Processing Structure Time=System Event Cycle=Processing Cycle	e.g., Knowledge Architecture End=Criterion Means=Option
Builder View	e.g., Data Design Entity=Segment/Row Relationship=Pointer/ Key	e.g., Structure Chart Funct=Computer Funct Arg=Screen/Device Formats	e.g., System Architecture Node=Hardware/ System Software Link=Line Specification	e.g., Human/ Technology Interface Agent=User Work=Job	e.g., Control Structure Time=Execute Cycle=Component Cycle	e.g., Knowledge Design End=Condition Means=Action
Subcontractor View	e.g., Data Definition Description Ent=Fields Rel=Addresses	e.g., Program Funct=Language Stmts Arg=Control Blocks	e.g., Network Architecture Node=Addresses Link=Protocols	e.g., Security Architecture Agent=Identity Work=Transaction	e.g., Timing Definition Time=Interrupt Cycle=Machine Cycle	e.g., Knowledge Definition End=Subcondition Means=Step
Functioning Enterprise						

Figure 11 - Zachman Framework (Zachman)

DoD Architecture Framework (DoDAF)

The DoD Architecture Framework borrows much from Zachman. DoDAF only has three views: Operational, Systems, and Technical Standard Views. Each of these views can be modeled and decomposed independently, but in order to maintain an integrated architecture, should be integrally linked to each other. The views are broken into products that facilitate the modeling and decomposition, as well as the view integration.

Currently the DoDAF treats security as any other requirement, and there is no separate "Security View." Many of the artifacts (nodes, interfaces, systems) have security requirements. Some of the items decomposed are actually security implementations, like Authentication. Others cannot function with out security.

DoDAF allows the use of both structured Techniques and Object Oriented to accomplish the development of the products. Many of the views can use Unified Modeling Language (UML) to decompose activities and processes. The Operational view describes tasks and activities need to perform the mission. It depicts the participating logical nodes, and the associated information exchanges (DoD, 2004
). The Systems view describes the systems and interconnections in context with the operational view (DoD, 2004
). The Technical Standards View (TV) is a profile of the minimum set of time-phased standards and governing rules for the implementation, arrangement, interaction, and interdependence of the systems involved (DoD, 2004
).

Other Frameworks

Two of the primary IT Frameworks in use today include the Java 2 Enterprise Edition (J2EE) and Microsoft .NET Framework. While their developments differ greatly, they do share many similarities. For example, from a security perspective, both implement a secure space for running code. In .NET it is called the common language runtime. In Java, the secure space was originally only in the "sandbox" of the Java Virtual Machine. Both implement role-based access control for resources. Other frameworks include the International Standards Organization Open Distributed Processing (ISO ODP) and The Open Group Architecture Framework (TOGAF).

Security Architectures

DGSA

DoD Goal Security Architecture (DGSA) was an effort for incorporation of security concepts into current and new DOD information system architectures [59]. It was created as a security architecture covering the full range of DoD missions and related information system security services and security mechanisms. From a modeling perspective, DGSA described four types of architectural "views"

- Abstract Architecture defines the high-level principles, concepts and functions to satisfy typical security requirements
- Generic Architecture describes the general solution and relationship of components
- Logical Architecture defines the design solution. It provides the blueprint for the actual specific solution
- Specific Architecture address the actual components, interface

These were part of an overall DGSA transition strategy which included several critical segments: The segments were: standards, product development, research and technology, security

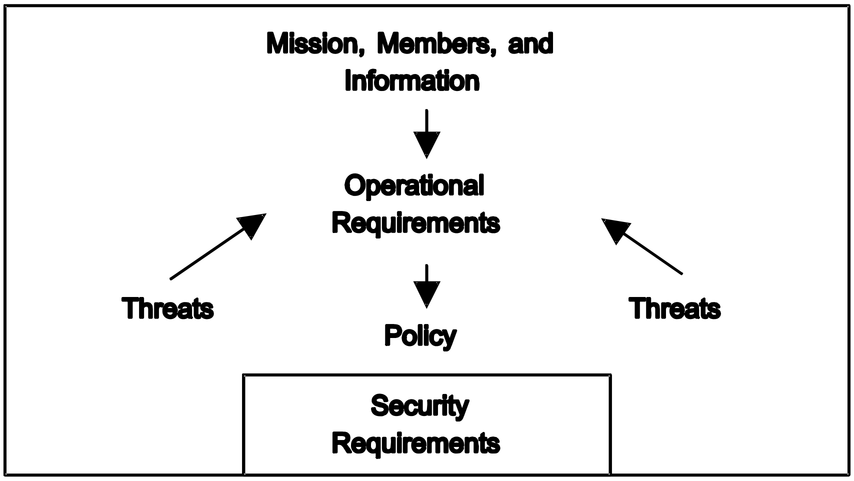


Figure 12 - DGSA Security Requirements process (DoD, 1993)

management, local subscriber environments, communications systems, certification and accreditation, policy and doctrine, and education and training.

shows how the DGSA succinctly described the establishment of security requirements using this simple mandatory process:

1. the information to be managed is identified
2. the operational requirements for the use of the information are stated
3. the value of the information is determined
4. the potential threats to the information are identified
5. the security policy can next be stated for the protection for the information based on the potential threats and/or the security services that afford the appropriate protection of the information based upon the value of the information and the threats to it

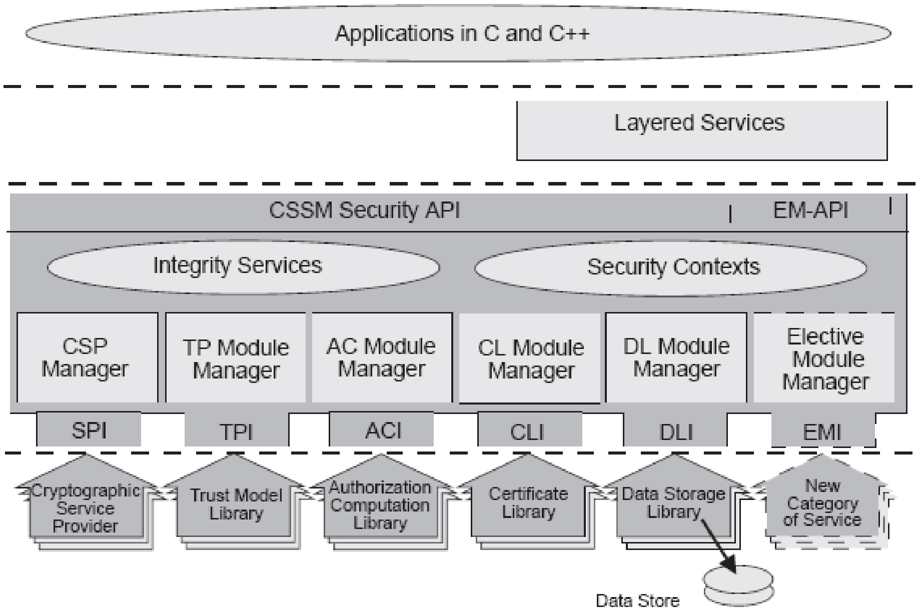


Figure 13 - Common Data Security Architecture (CDSA) (2000)

CDSA

In 1997, the Open Management Group (OMG) published the Common Data Security Architecture (CDSA), a set of layered security services (2000). Five architectural principles guided the CDSA development:

- Layered services
- Open architecture based on standards
- Modular and extensible components within layers
- Value in managing the details – CDSA components will handle the security details
- Emerging Technologies, such as portable tokens and digital certificates

shows the components and layers of the architecture. The Common Security Services Manager (CSSM), its modules and interfaces, is the core of CDSA.

NCOW RM (GIG Network Services)

Other relevant DoD Architectures regarding security are the Global information grid (GIG) Version 2 and its associated network services companion, Network Centric Operations and Warfare (NCOW) reference Model. The latest draft (Version 1.1) was to address information assurance, however this was deferred until the DoD IA Strategy is more fully developed.

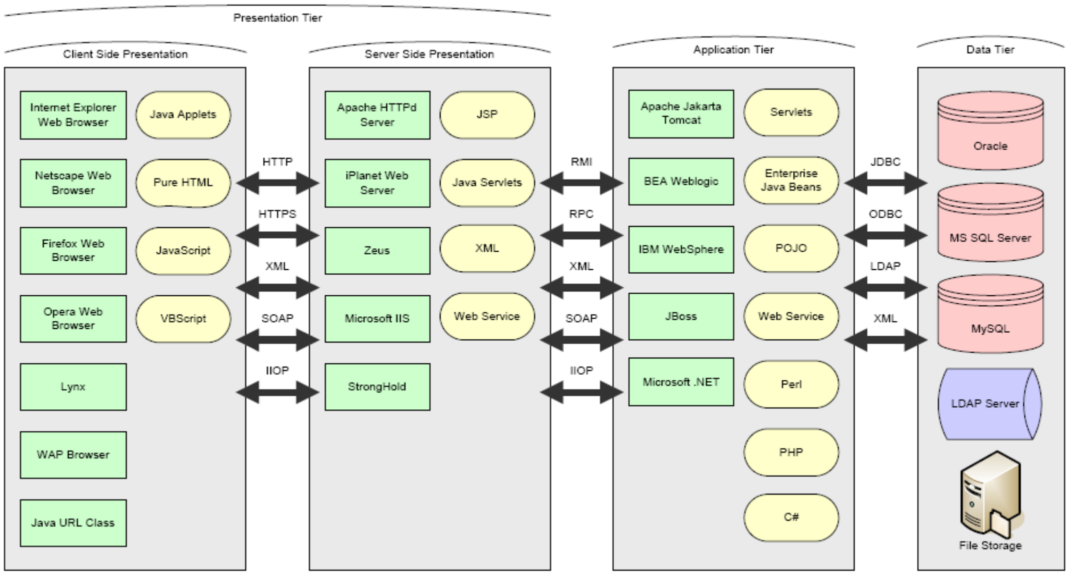


Figure 14 - Enterprise Information System []

As can be seen, there are many different ways to describe an Enterprise Architecture. Each is tailored to a particular environment. The crux of the problem is that there is no way to directly translate components of one to another. There is no overarching component to ensure there are no weak links between architectures. There is no common language among the frameworks and architectures, so that they can be used in conjunction to provide a secure interconnected combined infrastructure. When a Enterprise chooses to use a particular framework or architecture, they are locking themselves into that particular paradigm for the foreseeable future, since it will be cost prohibitive to move to a different one at a later time. Of course there are standards and protocols that will enable Enterprises with different architectures to communicate and share information, but not without cost. One of those costs is invariantly reduction of security, since the interconnected architectures typically are not designed to ensure security across architectures and it has to be added in later.

Models of Security

In an article by Edgar H. Sibley, Michael and Sandhu (Sibley et al., 1991), the authors state that all security policies must be explicitly defined. For example, all system should have a first fundamental policy:

There are objects whose contents should only be disclosed to or changed by authorized people, and these people should not disclose the details (or maybe even the existence) of an object to others who are not authorized to work with the objects. The objects should be preserved over time, if necessary for satisfactory continued operation.

Recent authors (Deng et al., 2003) attempt to model constraints or policies across the system or within components in a constraint language, such as Object Constraint Language (OCL). Security constraints may form well-defined patterns also. Thus, the ability to define a policy or constraint, as well as the analysis of that constraint across a system to ensure logical behavior should be present in security architecture.

Typically, most "architectures" are like those shown by Thomas and Sandhu [56], here describing database management systems scheduling the reading and writing across multiple security levels. They first describe the physical components (DBMS, OS, Trusted or untrusted front-end equipment and users), demonstrated in Figure 15a. Continuing in their article, they discuss the policies or system constraints. Showing message and process relationships, they then depict a pseudo activity model, shown in Figure 15b.

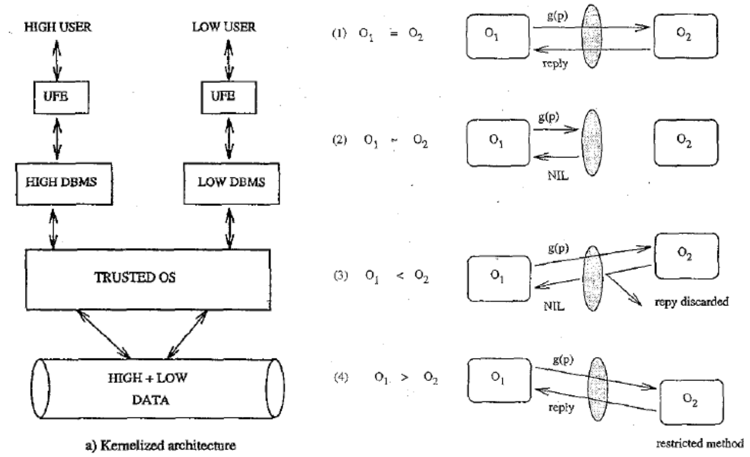


Figure 15 - a. Typical Physical Architecture and b. Typical Message (data) exchanges between two operations (process) at various relative security levels (Thomas and Sandhu, 1996)

Colored Petri Nets

As discussed earlier, the nature of security is dynamic. One way to model this behavior is to use Petri-nets (P-nets) or Colored Petri Nets (CP-nets or CPNs). CPNs are an improvement over their predecessors (P-nets) in that they allow the use of tokens that can carry data values. The token types or "Colors" used represent the type of data the tokens carry 0. CPNs provide a framework for the design, specification, validation, and verification of systems 0. The fundamentals of CPN are discussed in detail in (Jensen, 1998a, Jensen, 1998b, Kristensen et al., 1998). CPNs consist of Places (circles or ellipses), Transitions (rectangles), and Arcs (arrows) ().

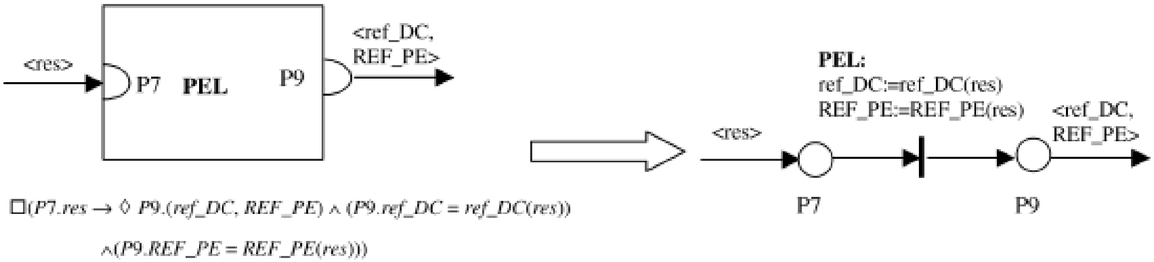


Figure 16 - Constraint PetriNet model, shown for a Policy Evaluator Locator (PEL) component (Deng et al., 2003)

Arch expressions describe how the state of the CPN changes when a transition occurs. Tokens are the data items at each place. The utility of CPNs is that they can be readily automated and enable behavior verification, performance evaluation, and state space analysis.

Different types of Petri-nets have been used to describe and analyze many types of security systems and security system components. For example, (Lim et al., 2001) used timed Petri-nets to model how to recover the private-key of a receiver and session key, when the receiver of a message in a cipher-text system loses its private-key. Fuzzy reasoning Petri nets are used in (Gao and Zhou, 2003) for intrusion detection and an example of how this concept can be used to detect a TCP SYN flooding attack is presented. Petri-net semantics of a Security Protocol Language is explored in (Bouroulet et al., 2004) and applied to the Needham-Schroder protocol to the violation of the authentication property. A Multi-level security model for multimedia document based on time augmented CPNs is proposed by (Joshi and Ghafoor, 2000), which allows for multiple security policy handling and hierarchical and path-based protection schemes. Another (Minami and Kotz, 2005) suggests a secure context-sensitive authorization system that protects confidential information in rules in a distributed environment modeled with Petri-nets.

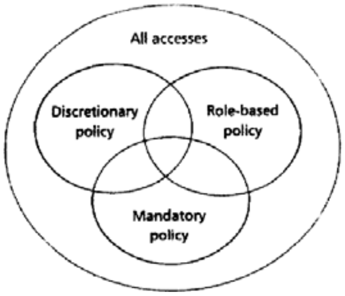


Figure 17 - Multiple Access Control Policies (Sandhu and Samarati, 1994)

Access control mechanisms using Petri-nets to model conditions and events are described in (Fugini and Martella, 1988) and are demonstrated for complex control polices building on discrete mechanisms implementing simple policies. The three types of access control policies are depicted in Figure 17. Discretionary Access Control policy control access to information based on user identity and authorizations, and is typical in most operating systems today (Windows XP, UNIX, etc). Role-based Access Control help enforce policies of least privilege and separation of duties by providing access based on the role the user is assuming at any given point while using the system. Finally, MAC policies are used to enforce information flow rules within a system. MAC enforces security classifications and need-to-know rules (i.e.: You must have a Secret or above clearance to access a Secret document, you must have a Top Secret (TS)/SCI clearance to access TS/SCI files, etc). MAC policies can be used to implement secure information flow within a lattice system ensuring the *-property ("no reads up/no writes down") paradigm as depicted in Figure 18 (Sandhu and Samarati, 1994, Bell and LaPadula, 1976). The use of Secure CPNs (SCPNs) to model an Enterprise's Mandatory Access Control (MAC) policies is proposed in (Juszczyszyn, 2003). The MAC model is further explored in (Jiang et al., 2004) building upon (Juszczyszyn, 2003, Bell and LaPadula, 1976) to provide a graphical analysis method suitable for formal verification.

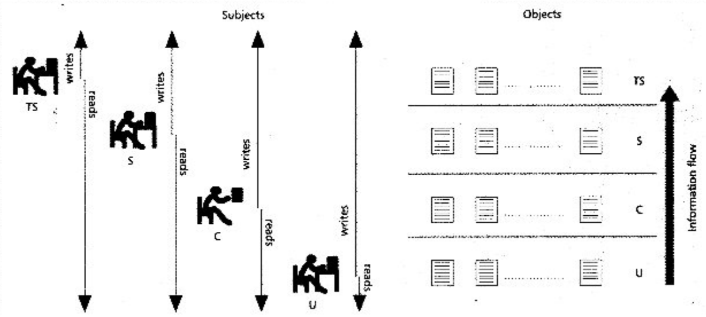


Figure 18 - Controlling Information Flow between Security Levels: Subjects can only Read Down and Write Up. (Sandhu and Samarati, 1994)

There are many examples of the use of CPNs in industry, for many diverse applications. One example is the use of CPNs to model a proxy firewall described in (Lee et al., 2003). Lee uses CPNs to model a complex proxy firewall, and then analyzes the design for stability, conducting a proof to that effect. CPNs were used to model the attack of the Slammer worm, and countermeasures against it (Stephenson, 2003). Stephenson also used CPNs to analyze root causes of incidents (i.e.: virus and worm attacks) (Stephenson, 2004a, Stephenson, 2004b)

Attack Trees

Attack trees, also known as AND/OR Trees, are a visual means of breaking down a task by goals and sub-goals to evaluate and compare threats (Schneier, 2000). The root node is the overall goal of the attack. Nodes at all levels below the root represent ways of achieving the overall goal by supporting the goal of the level above. Sometimes a node represents a standalone way of achieving the goal of the level above it, and sometimes one or more sub-goals are needed to achieve the goal of the level above. For instance, in Figure 19 the attack tree on the left only needs one of the leafs (Leaf 1 OR (Leaf 2 OR Leaf 3)) to satisfy the goal, whereas all three leafs of the tree on the right must be satisfied to meet the goal (Leaf 1 AND (Leaf 2 AND Leaf 3)).

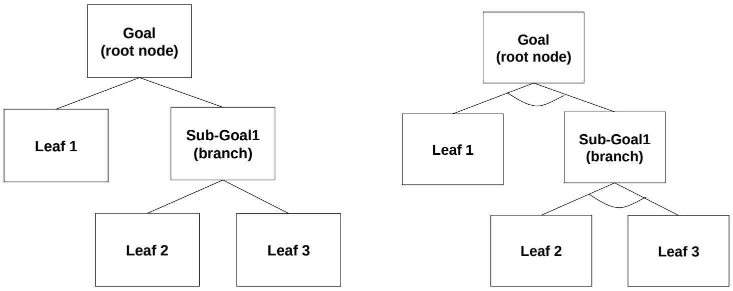
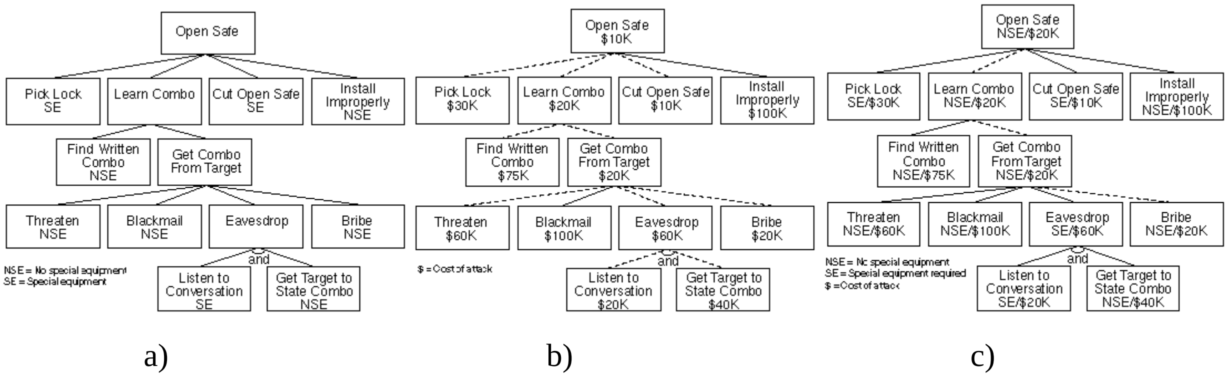


Figure 19- Attack Tree Conceptual View (Pallos, 2004). Attack Tree with ORs (left). Attack Tree with ANDs (right).

The textual view of these two trees would be (Pallos, 2004):

Left tree	Right tree
1. Goal	1. Goal
1.1. Leaf 1 (OR)	1.1. Leaf 1 (AND)
1.2. Sub-Goal 1	1.2. Sub-Goal 1
1.2.1. Leaf 2 (OR)	1.2.1. Leaf2 (AND)
1.2.2. Leaf 3	1.2.2. Leaf 3

Each of the nodes can be assigned a value. This value can be nominal (i.e.: possible/impossible, special equipment needed/not needed) or ordinal (i.e.: time of attack, likelihood of attack or monetary costs). Once values are assigned to the leaf nodes, the security of the goal (or against the goal) can be calculated. Multiple values can then be assigned to the various nodes and a composite value assigned and compared. For example, Figure 20a is coded with No Special Equipment (NSE) or Special Equipment (SE) needed in order to open the safe. Notice also that this tree has not been "calculated." The next tree, Figure 20b, is coded with the cost for each node and has been calculated with the lower cost taking precedence and rolled up to the top node. The combination tree, Figure 20c, is a combination of the two previous trees, and has been calculated, with lowest cost and NSE taking precedence. Given those constraints, bribing someone for the combination of the safe is the best solution to opening the safe.



a) b) c)

Figure 20 - Example of Combining Multiple Values a) and b) onto a Single Attack Tree, c). Leaves are assigned different "values" of a possible attack. Security of Goal is "calculated" using Boolean (AND/OR) operations. Dashed lines represent attacks possible under given conditions (i.e: Cheapest attack with no special equipment). (Schneier, 2000, Schneier, 1999)

Unifying Theme – Securing the Enterprise

One of the most common unifying themes that come to mind when thinking about enterprise security is *Technology, Policy, and Procedures.* Unless all three are applied in a systematic and compressive way, enterprise security will suffer. DoDAF, as well as the other frameworks, enables the modeling of all three of these tenets of enterprise security; however there is still room for improvement. This is not a criticism of DoDAF or the other frameworks, but rather an observation of areas requiring more attention. One area that needs more attention is attack modeling and response with respect to enterprise security. There are many different ways to approach this, and many are in use today. Tying together what has been done before is a challenge. There are many examples of implemented systems that are extremely robust and apply comprehensive security policies and procedures, however when we begin to tie these systems together with other systems failures occur due to "weakest link" or interface security flaws. Moreover, since threats are continually evolving, we need our enterprise systems to proactively counter these threats (whether known or unknown). In order to satisfy security requirements for an enterprise, we need an Enterprise Architecture that integrates security fundamental logical principles, formal logic, policy, threats, system components, standards and other miscellaneous artifacts. If security is not implemented across an Enterprise or Community of Interest in a consistent, methodically planned manner, global vulnerabilities may be uncovered due to local deficiencies. When we talk about net-centric operations, we're not simply referring to only business applications like Excel or PowerPoint. Rather, we imply control, health, status and operation of military weapon systems on the network. Therefore, we must model security as an integral feature of the overall architecture, with the foreknowledge that security is only as strong as the weakest link. An Enterprise Architecture should provide a blueprint on how to fit all of the fundamental components of the Enterprise IT systems together, and will be a tool for decision makers and procurement officials.

Most importantly, all aspects of the enterprise architecture must be integrated. Levis et al [41, 42, 43] refer to this as concordance. When that is accomplished the true effects of any change on the rest of the architecture are readily apparent, and intelligent decisions and trade-offs can be made. The DoD framework (DoD, 2004
) states that "Integrated architectures provide a logical, structured approach for defining how forces operate, the associated information flow, the relation between that information flow and system capabilities, and the relation between system capabilities and technical standards."

Since security has both static and dynamic components, and system boundaries can change during runtime, the modeling performed must be both static and dynamic. Executable architectures stochastic models, and other tools and methods must be developed to help determine the best methods for securing our enterprise. That's where Petri nets, Attack Trees, and other type of modeling tools may prove useful. As suggested in Figure 21 there is a gap between low level models and various architectures, especially in terms of modeling security. The problem with most models, architectures, and frameworks is that there is no direct correspondence between the various levels, and they are usually not consistent with implementations. Many times models and architectures are developed and are relegated to shelves and never looked at again. With the foreseeable future of continual changes the problem arises of how to keep the architecture current and relevant without it becoming stale or overly complex/obfuscated. What may be needed is a top to bottom system of models that can be used to forecast results of planned changes to an Enterprise's infrastructure. This architecture not only has to be secure but optimized for the particular business. A generic system of models can therefore be developed that is then tailored for the individual enterprise type (or specific enterprise).

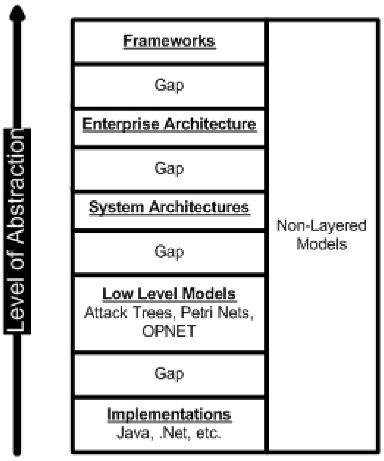


Figure 21 –Hierarchy of models with modeling gaps identified.

Arguably the most important aspect of the Enterprise Architecture is how well it captures the business practices of the enterprise and how adaptive to changes as these practices evolve. That it to say, how well the architecture facilitates the enterprise's current mission and how easily it can be adapted to new technologies, polices, and procedures. To be truly useful we may find that the enterprise architecture must be performance based, executable, and adaptive. Much like how a wind tunnel is used to develop aircraft, an enterprise architecture, built from a hierarchy of models, must be used to "fly" changing technologies, polices, and procedures before building the architecture and finding out it's not suitable (but being force to accept it because of fiscal restraints). By building a strong foundation of underlying models, the enterprise can tweak configurations, accept or reject component, and play "what if" games to create the best infrastructure for that particular organization. From this an *enterprise benchmark* can be created, that can be used to measure the effect of configuration or policy change (or any change in technology, policy, or procedure) on rest of enterprise. Even more exciting is that the enterprise could then simulate attacks and component failures, determine their effect, and develop courses of action to mitigate the attack and normalize operations as quick as possible.

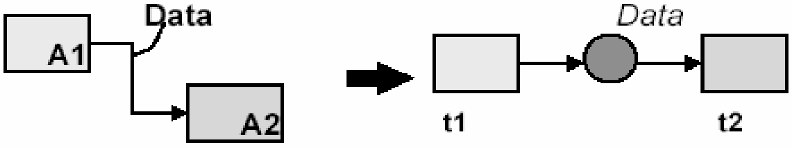


Figure 22 – Direct mapping from an activity model to a Petri Net []

So, for example, we can begin this hierarchy of models, much like eating the elephant – one bite at a time. We can develop a top down modeling methodology, whereas the operational and system activities are decomposed, attacks and protections are modeled against these activities, and the entire enterprise model implemented in a way that facilitates execution. One way would be to build Petri Net models from the activities, as shown in Figure 22.

While DoDAF is good at mapping operational activities to systems, security threats/attacks or their countermeasures are not easily captured or modeled using this framework. There has been some effort to integrate security into the activity models, but it seems the effort has been attempting to isolate security features into activities, which is a challenge, especially when it seems that many of the activities themselves are enabled or protected by security mechanisms. For instance, how do you model confidentiality, when it underlies activities of almost every other activity.

Today the focus on incident and threat handling is primarily applying patches, tightening down individual implementation configurations, etc. For a NetCentric system more will be required. Combatant Commanders and CIOs will need tools to couch enterprise security decisions in terms of operational mission. Specifically they'll need the capability to develop a "courses of action" infrastructure to handle security events. To get there we will have to develop a process that evaluates security events and produce a list of courses of action (COA). Some of these COAs will require automatically response in certain situations and require human intervention for others. For instance, no sane person would ever want a computer to automatically decide to release a nuclear weapon. This policy decision would have to be built into (as is) into a defense enterprise's infrastructure (technology, policy, and procedures). However, for an e-commerce company it may make sense to automatically adjust system configurations (i.e.: restrict certain types of network traffic, require more thorough authentication, etc) based on a threat warning, since threats can propagate through the internet too rapidly for a human response. A business decision would have to be made as to what would be restricted and to what degree based on solid risk management principles after performing some type of cost/benefit analysis.

To do this we must distinguish between strategic and tactical decisions, understanding that tactical decisions must be based on strategic vision. This is especially true for NCOW to be successful, but applies to the business world as well; only the ramifications are more devastating. We need a broad strategic view when making these tactical decisions. For example, when vulnerabilities were discovered in Active X the DoD response was to restrict its use. However, the enterprise suffered because applications using Active X were not available. Now say a Commander/CIO knew that shutting off Active X would take down the Logistics System, she/he would probably look for a different solution. In this light we really need a mapping of all hardware and software components in an enterprise related to mission areas and an analysis system to take the large volume of information and produce a time sensitive course of action list. Answers to questions like, "What do we lose if power goes out in the East Coast?" or "How do we respond to a Distributed Denial of Service attack against our web server?" need to be answered before they happen. Today it is difficult to plan for all contingencies, but by developing adaptive architectures based on models that can simulate attacks and failures; we will be better able to do so.

In order to build adaptive architectures we will first need to capture and model the process of adding new components into an enterprise. A simplistic Enterprise-System Level interaction model is depicted in Figure 23. When new components are added to an enterprise, not only are the new capabilities it offers added, but also new vulnerabilities. There is also the likelihood that some of the interactions between the new component and the existing systems may not be fully understood or desired. The enterprise provides tools, architecture, and policy to minimize the undesirable interactions, and the legacy system itself can assess changes made by the new additions. It does this primarily by enforcing constraints on the systems and subsystems that it is composed. Changing threats may require changes to the enterprises tools, architecture, and policy and may result in guiding changes to system level components. These new threats may result in further system-level constraints being levied.

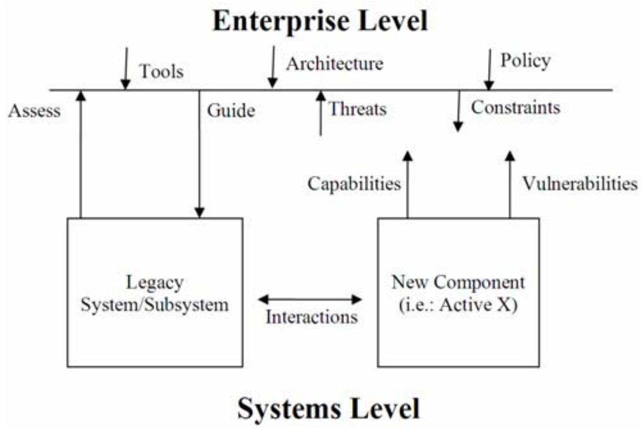


Figure 23 – Enterprise-System Level interaction model. Dynamics of adding new systems level components to an Enterprise

It's been said, "A journey of a thousand miles starts with a single step." We are at the dawn of the information age, and our first steps must include a secure framework around which we can build the information enterprises of the future.

References

Merriam-Webster Online Dictionary. Merriam-Webster

(2000) Technical Standard, Common Security: CDSA and CSSM, Version 2.3. The Open Group.

ALBERTS, D. S., GARSTKA, J. J. & STEIN, F. P. (2003) *NETWORK CENTRIC WARFARE: Developing and Leveraging Information Superiority*, C4ISR Cooperative Research Program (CCRP).

ALBERTS, D. S. & HAYES, R. E. (2006) *UNDERSTANDING COMMAND AND CONTROL,* Washington, D.C., DoD Command and Control Research Program.

BELL, D. E. & LAPADULA, L. J. (1976) Secure computer system: Unified exposition and Multics interpretation. The MITRE Corporation.

BOUROULET, R., KLAUDEL, H., PELZ, E., KISHINEVSKY, M., DARONDEAU, P., KISHINEVSKY, M. & DARONDEAU, P. (2004) A semantics of Security Protocol Language (SPL) using a class of composable high-level Petri nets. *Proceedings. Fourth International Conference on Application of Concurrency to System Design.* Los Alamitos, CA, IEEE Comput. Soc, Software Quality Res. Lab., McMaster Univ.

DENG, Y., WANG, J., TSAI, J. J. P. & BEZNOSOV, K. (2003) An approach for modeling and analysis of security system architectures. *IEEE Transactions on Knowledge and Data Engineering,* 15**,** 1099-1119.

DOD (1993) Department of Defense (DoD) Goal Security Architecture (DGSA) Version 1.0. Defense Information Systems Security Program.

DOD (2004) DoD Architecture Framework (DoDAF) Version 1.0

ERL, T. (2005) *Service-Oriented Architecture: Concepts, Technology, and Design*, Prentice Hall.

FUGINI, M. G. & MARTELLA, G. (1988) A Petri-net model of access control mechanisms. *Information Systems,* 13**,** 53-63.

GAO, M. & ZHOU, M. (2003) Fuzzy intrusion detection based on fuzzy reasoning Petri nets. *Systems, Man and Cybernetics, 2003. IEEE International Conference on*

IDA (2004) Security Activity Budget Estimates Institute for Defense Analyses.

JENSEN, K. (1998a) A brief introduction to colored Petri nets. *Proc. Workshop on the Applicability of Formal Models.* Aarhus, Denmark.

JENSEN, K. (1998b) An Introduction to the Practical Use of Coloured Petri Nets. *In: W. Reisig and G. Rozenberg (eds.): Lectures on Petri Nets II: Applications, Lecture Notes in Computer Science,* vol. 1492**,** 237-292.

JIANG, Y., CHUANG, L., ZHEN, C., HAO, Y., MEMON, A. M. & MEMON, A. M. (2004) Using Petri nets to verify access policies in mandatory access control model. *Proceedings of the 2004 IEEE Conference on Information Reuse and Integration* Piscataway, NJ.

JOSHI, J. & GHAFOOR, A. (2000) A Petri-net based multilevel security specification model for multimedia documents. *IEEE International Conference on Multi-Media and Expo.*

JUSZCZYSZYN, K. (2003) Verifying enterprise's mandatory access control policies with coloured Petri nets. *Proceedings of the Twelfth IEEE Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises.*

KRISTENSEN, L. M., CHRISTENSEN, S. & JENSEN, K. (1998) The practitioner's guide to coloured Petri nets. *International Journal on Software Tools for Technology Transfer,* 2**,** 98-132.

LEE, M.-K., ARABNIA, H. R. & MUN, Y. (2003) Stability verification of proxy firewall using coloured Petri nets. *International Conference on Security and Management.* Athens, GA, CSREA Press.

LIM, S.-Y., KO, J.-H., JUN, E.-A. & LEE, G.-S. (2001) Specification and analysis of n-way key recovery system by Extended Cryptographic Timed Petri Net. *Journal of Systems and Software,* 58**,** 93-106.

LUCYSHYN, W. & RICHARDSON., R. (2005) 2005 CSI/FBI COMPUTER CRIME AND SECURITY SURVEY. Computer Security Institute.

MINAMI, K. & KOTZ, D. (2005) Secure Context-Sensitive Authorization. *Pervasive Computing and Communications, 2005. PerCom 2005. Third IEEE International Conference on*

PALLOS, M. S. (2004) Attack Trees: It's a Jungle Out There – Seeing Your System through a Hacker's Eyes. *WebSphere Online Journal,* 3.

RADUEGE, L. G. H. D. (2004) Transforming the GIG. *SPACECOM.* Colorado Springs, CO, Defense Information Systems Agency.

SANDHU, R. S. & SAMARATI, P. (1994) Access control: principle and practice. *Communications Magazine, IEEE,* 32**,** 40-48.

SCHNEIER, B. (1999) Modeling Security Threats. *Dr. Dobbs Journal.*

SCHNEIER, B. (2000) Secrets & Lies: Digital Security in a Networked World. New York, John Wiley & Sons.

SIBLEY, E. H., MICHAEL, J. B. & SANDHU, R. S. (1991) A case-study of security policy for manual and automated systems.

STEPHENSON, P. (2003) Modeling of Post-Incident Root Cause Analysis. *International Journal of Digital Evidence,* 2.

STEPHENSON, P. (2004a) The application of Formal Methods to Root Cause Analysis of Digital Incidents. *International Journal of Digital Evidence,* 3.

STEPHENSON, P. (2004b) Expanding on the use of Colored Petri Nets. *Computer Fraud & Security***,** 17-20.

TAYLOR, E. G. (2004) Transformation to Net-Centric Ops. *SPACECOM.* Lincoln Laboratory, Massachusetts Institute of Technology.

THOMAS, T. K. & SANDHU, R. S. (1996) A Trusted Subject Architecture for Multilevel Secure object-Oriented Databases. *IEEE Transactions of Knowledge and Data Engineering,* Vol 8.

USAF (2003) Air Force Doctrine Document 1 (AFDD1) Air Force Basic Doctrine.

USAF (2005) Air Force Doctrine Document 2-5 (AFDD2-5) Information Operations.

ZACHMAN, J. A. Framework for Enterprise Architecture.