Security Architecture Approaches

Sarah Pramanik, Northrop Grumman

Abstract. There are multiple approaches to capturing security architecture information. Each framework and taxonomy has strengths and weaknesses that lend themselves to different tasks. The purpose of this paper is to provide information on some approaches with suggested improvements. The choice of architecture methodology can increase efficiency or diminish it. It is necessary to understand which methodology provides the most effective approach.

1. Introduction

Security architecture development is key in determining whether a system will have an adequate security posture. A good security architecture allows system security engineers to understand what mitigations are necessary and how they need to be placed into a system. It also allows for an understanding of information flows between elements of the system and flows out of the system. Without this type of information it is not possible to ensure that the risk level of a system has reduced to an adequate level. There are many taxonomies, frameworks and methodologies for creating security architectures, although there is not a standard required in the DoD.

There are several thoughts on approaching security architectures. The first approach is through adding security into a normal Enterprise Level architecture, which is most common in the DoD, due to the required use of the Department of Defense Architecture Framework (DoDAF) [1]. The second is to have a separate system security architecture [2]. Many of the papers on security architectures [3] define a specific architecture for a specific situation, and are not meant as a pattern on which to base the creation of a system security architecture. These approaches are difficult to work with in designing tactical systems. Designing security architectures for an unmanned aerial vehicle or a submarine is quite different than building security architectures for an office environment.

Even though the requirement sets are identical in some cases, the interpretation and resulting implementation may look radically different. In the Department of Defense there are a few documents that cover security requirements and frameworks, such as the DoD Instruction 8500.2 [4], Information Assurance Technical Framework (IATF) [5], and NIST SP 800-37 [6]. Each system will be designed according to a specific set of security requirements, but the requirement sets are the same whether it is an enterprise type system or a tactical system. The security engineer must understand the purpose of the requirement and the underlying security principles well enough to translate them into a security architecture that will protect unusual systems.

This paper will compare various methodologies commonly used to create security architectures and will provide suggestions on how to apply them for tactical systems. Many of the existing methodologies provide a solid basis for beginning an architecture, although others are sorely lacking. Choosing which methodology to follow is crucial, as it will be the basis on which a security architect will potentially have to work with for many years. The wrong framework can lead to a great headache for engineers and can lead to poor security postures.

2. Background

There are multiple frameworks, and taxonomies for the creation of system architectures. Some of these are specific to system security architectures and others are system oriented with security views. In either case, the system security architecture must be totally aligned with the system architecture. This can be especially difficult when designing a non-enterprise type system.

The security architecture is a detailed explanation of how the requirements need to fit within the system. This needs to be done in such a way as to ensure that all vulnerabilities are covered, and not just by one layer of defense, hence the heavy use of the term “defense in depth”. Many methods have been suggested [7, 8, 9, 10, 11, 12, 1, 13, 14, and 15]. Each has a similar goal: build security into the system as a part of the architecture. Many times security is bolted onto the end of system development, or as an upgrade after the system is completed. Each framework and taxonomy has a different way of representing the security elements of a system. One of the earliest taxonomies developed for building enterprise architectures is the Zachman Framework.

The Zachman Framework is used for organizing architectural artifacts [13]. This framework did not supply a formal methodology or process for collecting the artifacts, and so other frameworks were created from this base. The DoDAF was built off of this Framework. SALSA [9] and the Sherwood Applied Business Security Architecture (SABSA) [15] are both built off of the Zachman Framework as well. SABSA is one of the few frameworks developed to focus on the security aspects of a system. Its goal is different than that of the NIST Risk Management Framework (RMF). The intent of the RMF is to improve information security, strengthen risk management processes, and encourage reciprocity among federal agencies. Each of the Frameworks and Taxonomies has strengths and weaknesses.

The biggest complaint made in previous research is that security is done in an ad hoc manner [16]. If an overall security picture of the system is not developed in the beginning, technologies and procedures will be thrown at the system in hopes that something will stick. That leads to the hope that the stuff that sticks is good enough. How can you know what is good enough, without a strategic understanding of the system, its environment, its operations, users and the like. Simply, you cannot. It is not possible to do a risk assessment of a system, if all of the information flows, vulnerabilities, threats and mitigations are not known. The frameworks, taxonomies and methodologies all aim to layout this information.

2.1 DoDAF

The DoD requires the use of DoDAF for functional system architectures. The goal of the functional architecture is to show how different types of data should move through the various functions of the system. From this model the system engineers are to derive the interface requirements. This model is also supposed to help define the subsystems and allocation of requirements.

In looking at the DoDAF, each view takes a system and tries to explain it in a different fashion from the other views. There are operational views, system views, technical views, and all views [1]. These describe a system using specific techniques.

The operational views (OV) provide a very high level view. They pictorially depict how different pieces of the system should communicate. These flows do not show any component level information. They show the flows between various organizations, specifically activities, rules, states, events and interoperability flows.

The system views (SV) start to breakdown the flows into system level components. The flows show services, interfaces and data exchanges. This is also the view that is supposed to start to break down the system into a physical schema.

These views do not always show actual protocols. This means that if one segment is planning on using TCP/IP for connection and another is planning on UDP it will not necessarily be evident in this view. It truly depends on how an organization chooses to use the framework. This can lead to confusion between segments. Many times these details are left up to the individual designers of each segment. If there is no communication between the designers then the segments will not communicate.

The technical views (TV) explain the technology standards that should be used in the system, and the emerging standards. These can be difficult to pin down. A system cannot be on contract for an incomplete or unsigned standard. In a system where the development will take several years to complete, it is unknown when the project begins which draft documents will become standards. Typically, this means that if a program lasts for any substantial amount of time then the system will not be created using the newest standards, just those that are on contract.

The all views provide summary information and a glossary. These are helpful in defining the language in which the developers will use to communicate. This can be one of the most useful tools in the framework for developers. The system can only be created if all the designers, developers and architects can communicate clearly. The use of a system wide dictionary can help facilitate this communication.

In each of the views, it is possible to add security attributes. However, this does not adequately express all the security considerations. Security issues can easily be lost in the overall enterprise architecture. This is one of the reasons that, for some systems, it becomes important to break out the system security architecture out of the enterprise architecture and bring it into its own architecture. This also must be done carefully, to ensure that the security architecture aligns with the system architecture.

Another aspect that must be considered in an upcoming program is that DoD is moving from DIACAP to the RMF. This then means that a program will potentially be required to create security views in the DoDAF and follow the NIST RMF. Ensuring that these two methods do not disagree with each other and ensuring that inefficiencies are not increased will be a new task for the system security architects.

2.2 NIST RMF

The RMF is designed to allow security engineers to understand the risk level associated with a system. It is supposed to work in conjunction with normal systems engineering. The RMF in conjunction with the allocated information assurance controls from [6] provide something similar to the DoDI 8500.2 [4] and DoDI 8510.01 [17]. The wording in [6] is less vague than that in [4] and provides a more granular way of allocating controls. In and of itself it does not provide guidance on what the security architecture is to look like, nor what it should contain, and is similar to suggestions made in [18]. Its only requirement is that the controls are allocated in such a way as to reduce risk to an acceptable level. It is still up to the security architect to select a method of describing the architecture.

The RMF provides good guidance on boundary determinations as well as words of caution in dealing with SOAs [19]. It also provides a list of criteria that a system security architect should consider when drawing boundary lines and with respect to the overall risk of a system, as well as other information that security engineers would do well to heed. It is also meant to be cyclical.

Once controls are allocated, security engineers are to continually monitor how well they abate the risk and modify mitigations when necessary. Although this is a noble goal, it becomes more challenging if a security architect has not laid out the information in a way that is clear and understandable.

2.3 SABSA

The SABSA is a matrix based taxonomy very similar to the Zachman Framework, but with a security twist. Each layer of the SABSA matrix [20] is meant to force the security architect to ask questions to ensure they understand all the various attributes of security mitigations. Each layer takes a different view of the system and provides an explanation to a different set of developers and designers. It is also meant to provide a basis for conversation between the security team and all the other engineers. Although this model provides a good framework for asking questions, it does not match the model for DoD systems.

For example, in looking at the SABSA matrix, the facility manager's view is primarily focused on backups and handling support for what is supposed to be a fixed facility. Although this is a good start, there is a need to tweak this for DoD applications. Physical security is also something that must be kept in mind if dealing with both fixed and mobile platforms. Although physical security is discussed in [2] it assumes fixed facilities, in the DoD environment, the guns, guards, gates and dogs may not always be available to protect a mobile platform such as an Unmanned Aerial Vehicle (UAV). The issue of mobile and fixed platform must not only be part of the CONOPS and functional/conceptual architecture, but must also be folded into the detailed architectures as well.

2.4 IATF

The IATF is very different from the RMF and other frameworks. It is a living document that contains information on how to use specific security mitigations such as cross domain solutions and

firewalls. There are elements of the document that are not complete and leave room for new information as technology changes.

The IATF is best used in conjunction with the other frameworks to provide the security engineer better information on a particular mitigation than is provided in any of the other frameworks. In and of itself the IATF does not provide a discussion on how to go about putting a security architecture together, but provides useful information on various pieces that might be incorporated into an architecture.

3. Tasks

The initial task in creating a security architecture is outlining communication between components within a system and in and out of the system boundaries. This is true whether a system is an enterprise or a tactical system. This outline is similar to the system views in DoDAF, however the security architecture must define which of the flows will be classified (and what levels), sensitive or unclassified. Although the DoDAF allows for security attributes, these do not furnish enough information for the security engineers. This piece of the architecture will be built upon throughout the security architecture process.

Along with this, the security architect must understand the CONOPS and the intended use of each part of the system. This will outline the high level data flows for the system. There is a distressing tendency for system engineers to just throw all the security attributes into one function in the functional architecture. This is especially true if the security engineers are not part of the functional architecture team, or if security is bolted on at the end.

One of the unique aspects of tactical systems, unlike enterprise systems, is that portions of the system may be mobile. If working with an UAV, the architect must take into account not only the permanent ground station, but the UAV and in some cases mobile ground stations. The system may be connecting through differing networks at various classification levels. If the system is going through unprotected territory, there must be means of protecting that information, such as cryptography or the use of volatile memory. This flows both from the understanding of the flow of information as well as CONOPS.

The security architecture needs to have both functional and physical details which must meet the operational needs of the mission. These need to be tied together to ensure that the physical manifestation of the system performs the functions that are intended. These should include such things as key assets, the most critical pieces of information that need to be protected, user interfaces, unsafe environments, etc. Once these are identified, then the next step is to figure out how to mitigate or eliminate these vulnerabilities. This is described in the following approach.

3.1 A Modified Approach

The first step in creating the security architecture is to understand the requirements of the system. Aside from the information assurance requirements, such as the DoDI 8500.2 [4], the concept of operations, data classification and data exchanges must be recognized. This information should be translated into a series of block diagrams. Although a functional architecture must exist showing this information, it will not necessarily have to overlay on a physical diagram at the end. One thing to note is that as soon as the diagram goes under configuration control, it makes it difficult to maintain.

The first diagram should show internal and external flows of information similar to the one shown in Figure 1. The DoDAF OV-1s provide similar information from a functional standpoint. The security architect should ensure that the functional information matches the physical architecture from a security standpoint. The complexity of this view can increase significantly when dealing with multiple levels of security as opposed to operating a single security level.

Once the basic flows are understood, the classification level of each flow should be identified. There should also be notations of where data will cross a classification boundary, such as shown in Figure 2. This information allows the security architect to know where specific solutions such as encryption, cross domain solutions or data guards might need to be employed.

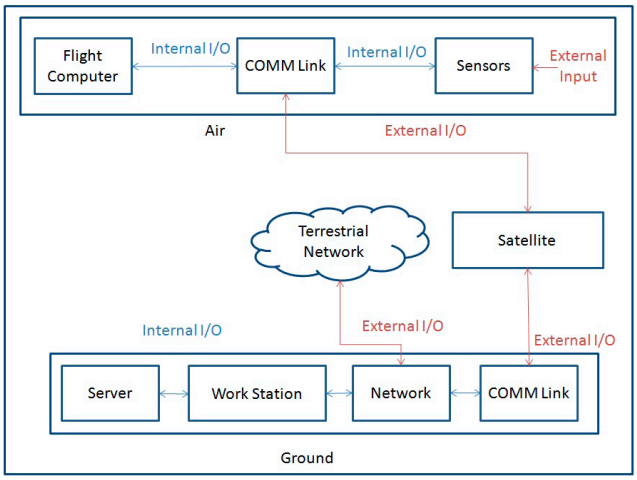

Figure 1. Flows of the System

As the DoDAF functional architecture is being completed, a corresponding physical architecture is typically being created, especially when the program is employing concurrent engineering. The physical architecture should be of great interest to the security architect. As the physical architecture is being designed, it will go through multiple iterations.

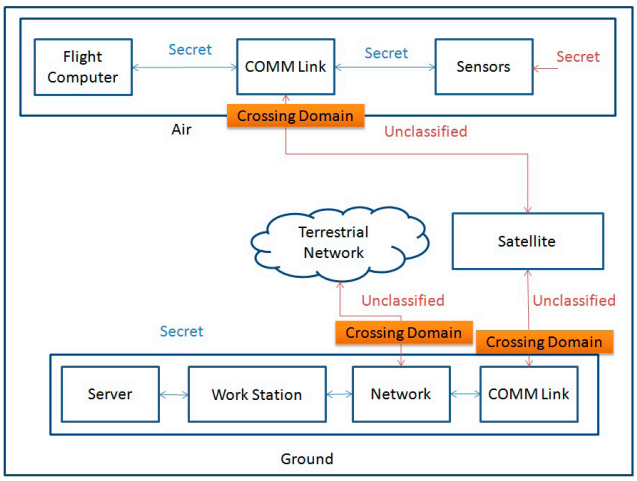

Figure 2. Classification Domains

As the first physical architecture is laid out, it becomes a base on which to build the physical security architecture. One of the primary purposes of the security architecture is to ensure that the security requirements are being met in the system. As subsystems and components are being defined the security engineers should begin listing the security requirements that will be applicable to each item. A spreadsheet can be maintained for each major component, in the beginning.

The first tab should map the IA controls to the parts of the component and to the expected mitigation, as seen in Figure 3. The engineer can include notes and justification for clarification. The second tab should map the IA control to the requirements associated with the control, as seen in Figure 4. The IA Control is used to link the mapping and the requirement tabs together.

This mapping allows for the security engineers to double check that requirement wording has been allocated appropriately. In the example, the security engineer should see this and note a need for a change in procurement specification wording. This is also an important part of the overall system security engineering method, because not all aspects of the security requirements can be laid out on a physical diagram. Requirements such as configuration management must be met for every component, but are more accurately addressed using language rather than diagrams. Later on in the lifecycle, the mapping can also be used to link it to verification of each of the controls.

	A	B	C	D
1	IA Control	Mitigation	Documentation/ Justification	Comment
2	PECS-2	Sanitization	Analysis Report - Appendix B Section 2.1.8	
3	DCSS-2	Must provide integrity controls, such as hashed OFPs, Authentication before changing an OFP, etc.	Analysis Report - Appendix A, Section 2.1.2	
4	IAIA-2	prevent the inclusion of embedded passwords	The Sensor contains a password to allow for loading the Operational Flight Program (OFP)	IAO agrees in principle, but needs to see indepth data flows, see email
5	ECSC-1	Hardening	This is a special purpose box, which uses the vxworks OS and does not contain anything that can be hardened using STIGs	Document X Rev B, Attachment 1, page 8, para 1.2.4 - They will harden the OS to the extent possible via vendor guidance, although not according to a STIG
6	DCPD-1	Free/open source software must be tested	The Sensor does not contain any FOSS/PD	This is N/A

Mapping | Requirements | Sheet3

Figure 3. IA Control Mapping

	B	C	D	E	F
1	IAC Tag	FRD/SDR Requirements	FRD/SDR Allocatio	SSS Requirements	HRS / SRS / Procurement Spec Requirements
2	COTR-1	**SDR_8976:** (U) The Sensor segment shall provide the capability to recover from a service discontinuity in a secure and verifiable manner (recovery without protection compromise)	Sensor 1	**Sensor1_19713:** (U) The Sensor shall provide the capability to recover from a service discontinuity in a secure and verifiable manner (recovery without protection compromise)	**Sensor1 Proc Spec_246:** (U) The Sensor shall conform to Department of Defense (DoD) and United States Navy (USN) security guidance regarding Information Assurance (IA) (DoD Directive 8500.1, DoD INST 8500.2, DoDI 8510.01, and CJCSI 6510.01) for a MAC1 classified system.
3	DCAS-1	**SDR_8984:** (U) The Sensor segment shall use IA or IA enabled COTS products selected from the approved NIAP products list.	Sensor 1	**Sensor1_19714:** (U) The Sensor shall use IA or IA enabled COTS products selected from the approved NIAP products list.	**Sensor1 Proc Spec_246:** (U) The Sensor shall conform to Department of Defense (DoD) and United States Navy (USN) security guidance regarding Information Assurance (IA) (DoD Directive 8500.1, DoD INST 8500.2, DoDI 8510.01, and CJCSI 6510.01) for a MAC1 classified system.
4	DCAS-1	**SDR_8993:** (U) The Sensor segment shall use IA or IA enabled GOTS products selected from the approved NIAP products list.	Sensor 1	**Sensor1_19715:** (U) The Sensor shall use IA or IA enabled GOTS products selected from the approved NIAP products list.	**Sensor1 Proc Spec_246:** (U) The Sensor shall conform to Department of Defense (DoD) and United States Navy (USN) security guidance regarding Information Assurance (IA) (DoD Directive 8500.1, DoD INST 8500.2, DoDI 8510.01, and CJCSI 6510.01) for a MAC1 classified system.

Figure 4. Requirement Mapping

Dynamic Object-Oriented Requirements System or a similar requirement repository under configuration control should be used to maintain the information once it has reached a baseline. The mappings will change as the security architecture matures, but in order to begin creating the security views of the system, the security architect must first understand the basic function and contribution of each subsystem and component.

There are IA concepts to consider in DoD systems, regardless of the IA requirement set. These can be defined by the security views as listed in Table 1 and will change over time, as components are added or changed in the system. Also, there are times when control of a subsystem function may move from one integrated product team to another. For example, boundary defense may originally be allocated to both the communications IPT and the ground segment IPT. As the system grows, the network piece may then be owned by the communications IPT. In the example above, this then would mean that the communications IPT now owns the responsibility for all boundary defense in the system. The requirement allocation would stay the same, and the diagram would stay the same, but the ground segment would claim inheritance for meeting boundary defense. This would then be added back into the requirements mapping discussed earlier.

Each view is to be overlaid onto the physical architecture. By layering these views, using a tool such as Microsoft™ Visio, the defense in depth picture can be seen. It also allows the architect to see where vulnerabilities may still exist. The earlier in the life cycle of a program that an accurate architecture can be created, the easier it is for security engineers to develop affordable security mitigations.

In some cases there may be multiple physical architectures. For example, if an unmanned vehicle and a ground station are being designed, creation of two physical architectures is necessary. They must show the interconnections between the two

Security View	Describes
Domains and Interconnections	- Interconnections with other enclaves or networks - Accreditation boundaries - Should show the classifications of data within the system - Points of red/black separation with respect to TEMPEST boundaries
Network Flow	- Boundary Defense - Data at Rest (indicate protection) - Data in Transit (indicate protection) - Network management data - Partitioning of User Interfaces and Storage - VoIP/Phone lines (indicate protection)

Cryptographic Usage	- Certificate Authority
	- Key Storage
	  o Asymmetric
	  o Symmetric
	- Interfaces to Electronic Key Management System (EKMS)
	- DS-101/102 lines
	- Key loading areas (at the box, or a harness, etc.)
	- Key zeroization paths (hardware, software paths)
	- Cryptographic type: Type-1, Type-2, etc.
	  o Embedded, separate hardware, etc.

Handoff (for unmanned systems)	- Outline control steps
	  o How control is acquired
	  o How control is passed
	  o Include authentication steps
	  o *Non-repudiation of Command and Control items are critical and should be indicated

Patch Management	- Indicate which pieces of equipment have software/firmware that will require patching - Indicate how patches will be downloaded, tested, protected and applied - *It should be noted that patches will require regression testing of system elements
Mobile Code	- Indicate pieces of the system potentially have access to mobile code - Level of mobile code - Protections against unwanted mobile code
Audit	- Indicate elements of the system with audit capability - What is audited(e.g. can only log user, but not time stamp) - Central audit repository
Backup/Recovery	- Location of recovery software(e.g. backup images in safe) - Location of data that has to be backed up - Location of backup repository - Timing of backups (weekly, daily, etc.) - *This provides useful information for the disaster recovery plan
Zeroization/ Sanitization	- Define memory element in every component, (volatile, non-volatile) - Type of memory - Classification of data in each memory type - Sanitization approach for NVM - Zeroization approach for Volatile memory - Associated analysis for components that plug into classified networks, but are claiming to remain unclassified at shutdown
System State Changes	- Identify when the system is considered classified, unclassified (or range of classification) - Should have a view for system running, powered off, and in-transit (if applicable)
IA Physical Security (Typically associated with facilities requirements)	- Status of platform (mobile, fixed, semi-fixed, etc.) - Fire suppression systems (type and placement) - Smoke detectors (placement and type) - Temperature/humidity controls (for humans and equipment) - Lighting (emergency) - Power (backup generators, failover period, etc) - *Depending on the system, these items may already be in place and a site survey will provide this data
Other Physical Security	If a new building is being constructed, a separate physical security architecture will have to be created. These items are not generally covered in IA documentation. For reference see [21].

Authentication	- Indicate how/where users will authenticate to the system
	  o Single sign on, role based access control, group accounts, etc
	- Indicate machine to machine authentication
	  o Note how the authentication is being achieved

Table 1. Security Views

pieces and any interconnections to outside networks that may be used to allow them to communicate. Developing security views for both architectures is also essential.

One of the difficulties is ensuring that as the physical architecture changes, the security architecture changes with it. This is especially true when the network diagrams and ports, protocols and services are in the process of being defined. These flows must be captured as part of the network view. They will determine how access control lists and firewall settings are configured in the final system.

Creating a security architecture for DoD systems requires the architect to be in line with both the information assurance control allocation and the required certification and accreditation process. This means that the architecture must show how the controls have been allocated, why they have been allocated and how it affects the system. This must be documented in a way that allows the security architect to provide justification for allocation of controls as well as an understanding of the security posture of the system if a control is not met or something in the system changes. This is fundamental to ensuring the security posture of the system remains at an acceptable level.

If the security architect defines all of the above views and maintains the requirements mappings, the evidence necessary for certification and accreditation is also available. The vulnerabilities in the system can be examined and either mitigated or justified. This approach uses a similar process as SABSA, but with a tactical focus and not an enterprise focus.

4. Conclusion

The use of the appropriate security architecture methodology, taxonomy or framework defines the degree of difficulty an architect will have in adequately describing the security functionality of a system. The modified approach provided in this paper allows for incorporating the tactical nature of many DoD systems into the equation. If followed, it also ensures that the security requirements are accounted for. This is critical to achieving certification and accreditation.

ABOUT THE AUTHOR

Sarah Pramanik is the lead for software information assurance (IA) on her current program. She is responsible for leading the development, integration and execution of IA activities within the software team. Sarah has worked in multiple areas of cyber security, including vulnerability and penetration testing, information system security engineering, cyber security training and security architecting. Sarah holds a Bachelors and Masters degree in Computer Science from The University of Colorado, Colorado Springs.

Northrop Grumman Aerospace Sector
600 Grumman Road West M/S Z21-025
Bethpage, NY 11714
Phone: 516-346-7737
E-mail: Sarah.Pramanik@ngc.com

REFERENCES

1. Wikipedia. Department of Defense Architecture Framework. 14 December 2010. <http://en.wikipedia.org/wiki/Department_of_Defense_Architecture_Framework>.
2. John Sherwood, et al. Enterprise Security Architecture: A Business-Driven Approach. San Francisco, CA: CMP Books, 2005.
3. Elkins, Asa, Jeffery Wilson and Denis Gracanin. "Security Issues in High Level Architecture Based Distributed Simulation." Proceedings of the 2001 Winter Simulation Conference. 2001 © WSC'01.
4. Department of Defense. "Department of Defense Instruction 8500.2: Information Assurance (IA) Implementation." February 6, 2003.
5. National Security Agency Information Assurance Solutions Technical Directors. "Information Assurance Technical Framework, Release 3.0." September 2000.
6. National Institute for Standards and Technology Special Publication 800-37 R1. "Guide for Applying the Risk Management Framework to Federal Information Systems: A Security Life Cycle Approach." Guide for Applying the Risk Management Framework to Federal Information Systems: A Security Life Cycle Approach. Gaithersbergm MD 20899-8930: Computer Security Division Information Technology Laboratory National Institute of Standars and Technology, February 2010.
7. Blackwell, Clive. "A Multi-layered Security Architecture for Modelling Complex Systems." CSIIRW'08 May 12-14. 2008 © ACM.
8. Susanto, Heru. and Fahad bin Muhaya. "Multimedia Information Security Architecture Framework." 2010 © IEEE.
9. Sherwood, John. "SALSA: A Method for Developing the Enterprise Security Architecture and Strategy." 1996 © Published by Elsevier Science Ltd. doi:10.1016/S0167-4048(97)83124-0, n.d.
10. Neilforoshan, Mohamad. "Network Security Architecture." © Consortium for Computing Sciences in Colleges, 2004.
11. National Information Assurance Glossary. "CNSS Instruction No. 4009." 26 April 2010.
12. Oda, Michelle, Huirong Fu and Ye Zhu. "Enterprise Information Security Architecture A Review of Frameworks, Methodology and Case Studies." 2009 © IEEE.
13. Wikipedia. Zachman Framework. 10 December 2010. <http://en.wikipedia.org/wiki/Zachman_Framework>.
14. Wikipedia. "Information Security." 10 December 2010. <http://en.wikipedia.org/wiki/Information_security>.
15. SABSA Ltd. The SABSA Method. n.d. <http://www.sabsa-institute.org/>.
16. Hartig, Hermann. "Security Architectures Revisited." EW 10 Proc. 10th Workshop on ACM SIGOPS European workshop. 2002 © ACM.
17. Department of Defense. "Department of Defense Instruction 8510.01: DoD Information Assurance Certification and Accreditation Process (DIACAP)." November 28, 2007.
18. Suhair Hafez Amer and John A. Hamilton, Jr. "Understanding Security Architecture." SpringSim (2008).
19. National Institute of Standards and Technology . "Special Publication 800-53 Revision 3, August 2009, includes updates as of 05-01-2010" Recommended Security Controls for Federal Information Systems and Organizations." Gaithersburg, MD, August 2009.
20. SABSA Ltd. 2010 ©SABSA. The SABSA Matrix. n.d. <http://www.sabsa.org/the-sabsa-method/the-sabsa-matrix.aspx>.
21. Department of the Army. "FM 3-19.30: Physical Security." 8 January 2001.