

# **S-TRAIS: A Method for Security Requirements Engineering Using a Standards-Based Network Security Reference Model**

**(Standards-Traceable Reference Architecture for Information Systems)**

Peter Stephenson  
Netigy Corporation, San Jose, CA  
And  
Oxford Brookes University,  
School of Computing and Mathematical Sciences, Oxford, UK

**Abstract:** *The development of security requirements for an information technology (IT) system implies some specific needs. For example, the requirements generally should be traceable to some underlying business driver. Additionally, in certain environments the requirements may need to be traceable to a standard or set of standards. We find several examples of this in the financial community. Soon the health care community will have similar requirements. When the security requirements for an IT system must be traceable to some standard or set of standards, there is an implied requirement for testing that system to determine its compliance with the standard(s).*

*The standards-based environment drives initial requirements engineering in that security requirements must be developed which allow the IT system under consideration to be bounded unambiguously and capable of submitting to a test regime which will yield results traceable directly to both business requirements and the governing standard(s). The requirement for placing an identifiable boundary around an IT system is relatively easy to meet when the system is a single device, operating system or application. When the IT system under consideration is a complex network, however, meeting that requirement is not as simple.*

*This paper describes a technique for bounding a network security abstraction into a single, definable IT system (or “reference security architecture”). This reference security architecture has a discrete set of security requirements capable of being defined clearly and unambiguously, traceable to business requirements and applicable standards, and capable of being tested and evaluated in accordance with those standards. For simplicity and generality we will use the Common Criteria as the “applicable standard(s)” although there is no reason why other, industry specific, standards could not be applied as well. The described technique currently is being tested on several networks of various sizes.*

---

## **1.0 Background and Problem Statement**

Over the years there have been many approaches to the evaluation of IT systems in accordance with some set or sets of security standards. Typically, the approach has been focused at the component level. The security specification and evaluation of a single device, application or operating environment has been relatively straight-forward. Although certain standards have been the objects of criticism for their approach, single focus systems, such as stand-alone computers, have traditionally been simple enough to be capable of submitting to a rather simplistic form of security evaluation. Requirements engineering for such easily definable systems is manageable because the systems themselves are clearly bounded and strictly defined.

Further, the concept that security requirements should be driven by business needs may be addressed with relative ease since a single IT component can be thought of as a response to a business requirement or set of business requirements. In this environment, complex systems, such as computers, which contain operating systems, communications protocols and applications, often have been approached as a collection of pieces.

## **1.1 PROBLEM STATEMENT**

Today's networks have far surpassed, in their complexity, the IT systems described above. Paradoxically, today's networks are, however, composed of many of those same components that have been handled individually in the past. Security requirements engineering, therefore, may consist of defining the business and standards drivers for a particular complex network and attempting to define and construct that network out of evaluated components.

This approach requires stringent definitions of permitted operating environments, protocols, applications, configurations and a variety of other generalized security requirements. Evaluation testing of the environment as a whole is very difficult since there is no overall security requirements definition for the network in its entirety. The usual approach has been to evaluate the components and assume the network. Security requirements engineering for the network, then, has tended to be focused on the network components and their required interactions.

The difficulty with assuming that, because the network's components have been stringently secured, the network is secure (or has met the overall business/standards requirements as applicable), is that any weakness in a component subjects the network (or some other of its components) to the risk of compromise. Such a weakness could be in the form of a configuration error that traces to the interaction between trusted components (or trusted and untrusted components) as opposed to the components themselves, or it could be in the form of an additional component unexpectedly being introduced into the network.

What we consider in this paper is an approach that views the network holistically, admits of component level weaknesses and develops system level countermeasures such that the network as a whole is secure, both from internal and external attacks. In order to engineer security requirements for such a system we first must:

- Clearly and unambiguously define the business drivers for the network
- Clearly and unambiguously define the evaluation requirements for the network
- Clearly and unambiguously define the boundaries for the network

Note that we do not need to (although there would be no harm if we did) create a network comprised solely of evaluated components if we can show that the network environment meets the above conditions. Also note that the network's boundaries may well include multiple domains. As we will see, those domains and their interaction is a key component of our approach.

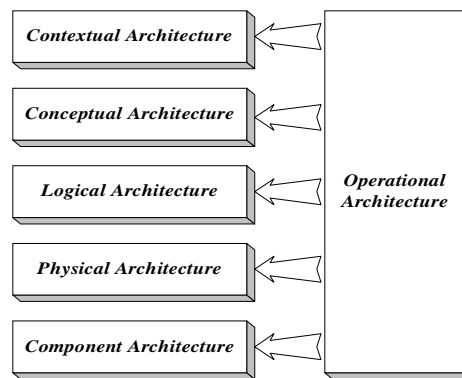
---

## **2.0 Prior Work**

Work on an approach that enables us to view a complex network as a single entity (or, at worst, a collection of definable domains) has been done by John Sherwood [SH99]. This paper describes the

SABSA methodology of creating a network security architecture using a reference model developed by Sherwood. The model may variously be depicted with five or six layers, the five layer version being the most instructive (shown below in Figure 2-1).

The SABSA model itself underlies Sherwood’s architecture methodology and gives it its power. However, the methodology is secondary to the role of the Model as an enabler of standards-based network evaluation. The unique aspect of the SABSA model is that it derives virtually all of its information directly from the business requirements for the network under development. It then depends upon the architect to apply some security approach, usually thought of as “best practices”, to address the Model’s well-ordered requirements. Because the concept of the original SABSA model has evolved over time, we will simply refer to the current instantiation of the model as the System Architecture Reference Model or SARM. We do not restrict this model to information security use since it can be applied effectively in other, non-security, information systems applications. This is, probably, the chief point of evolution away from the original SABSA thinking.



### System Architecture Reference Model

Figure 2-1

The layers of the Model including, the overarching *Operational Architecture* layer, are viewed as “architectures” Each architecture, for simplicity, is viewed from the perspective of the specialist most appropriate to make use of the information in the layer. For example:

- **Contextual Architecture:** The Business View
- **Conceptual Architecture:** The Architect’s View
- **Logical Architecture:** The Designer’s View
- **Physical Architecture:** The Builder’s View
- **Component Architecture:** The Tradesman’s View
- **Operational Architecture:** The Facilities Manager’s View

Each layer participates in the information gathering process in a standardized manner. Significant work on developing the processes associated with the Model has been done by David Lynas [LY99]. Lynas introduced the more formalized information gathering matrix shown in figure 2-2. Underlying Lynas’ depiction of the Model, and a significant contribution to the matrix, is work done by Zachman [ZA87]. The Zachman Architecture Framework is very similar to the Lynas matrix. However the Lynas matrix focuses upon information security issues while the Zachman framework is a more generalized view of any information system.

These two matrices are important for a number of reasons. For example, they allow the ordered gathering of information about the target system in such a manner as to focus on the business issues driving the system architecture requirements. Second, they permit the architect to gather those requirements in plain language instead of translating them into explicit technical representations. That translation occurs later in the process. Finally, they organize the collected information so that it may be used effectively later in the overall S-TR AIS process.

	<b>ASSETS</b>	<b>PROCESS</b>	<b>LOCATION</b>	<b>PEOPLE</b>	<b>TIME</b>	<b>MOTIVATION</b>
<b>CONTEXTUAL</b>	Business needs for information security	Business functions needing information security	Locations where the business is operated	Business security organization and relationships	Business security time dependency	Business goals and success factors
<b>CONCEPTUAL</b>	Business entities, relationships and information	Defense approaches	Security domains and security associations	Security authorities, organization and workflow	Time-related security concepts	Operational risk analysis and risk management
<b>LOGICAL</b>	Security-related entities, their relationships, and their logical representation	Security services	Security domain definitions and the security associations between them	Roles and privilege profiles for authorized entities	Security processing cycle	Security policies
<b>PHYSICAL</b>	Security-related data structure	Security mechanisms	Security technology infrastructure	Security user interface	Security execution control structures	Security rules, conditions and actions
<b>COMPONENT</b>	Security data field specifications	Security products, tools and standards	Security processing, addressing and protocols	User identities, privileges and access control lists	Security step timings and sequencing	Security procedures and steps
<b>OPERATIONAL</b>	Operational data security	Security operations and administration	Platform and network security	Operational support for users, operators and administrators	Security operations schedule	Business continuity plans

### SARM Information Gathering Matrix

Figure 2-2

It is important to note at this point that one of the underlying principles, when applying SARM to information security, is its dependence upon the application of security policy. In Section 3 of this paper we expand upon this important construct. Sweezy [SW00] discusses an approach that leads to the definition of security domains, or *policy domains*, based upon individual or groups of security policies. This, as we will see in Section 3, is not dissimilar to Lynas' approach although it is somewhat more stringently defined.

The SARM is the core of our approach. It allows us to define in considerable detail the requirements for a security architecture. There are, however, other approaches. For example, Ames et al [AM83] describe an architectural approach using security kernels. This approach, though popular, views the development of a security architecture from the perspective of the technology required to secure a system. The SARM on the other hand, views such an approach as a tool to meet a well-defined business requirement. While it has been pointed out that Ames intends that the security kernel approach be used in a number of systems (and, thus, can be a component in a larger system), SARM views the target system holistically and seeks to avoid the more granular approach of Ames.

Rita Summers, in her 1997 book, *Secure Computing* [SU97] describes the development of a secure system starting from a set of security requirements. This approach more closely follows the SARM model although the two approaches differ in some important ways.

## 2.1 THE COMMON CRITERIA AND OTHER STANDARDS

The history of the Common Criteria (ISO15408) is well enough known to allow us to skip it here. For the purposes of this paper, however, we must give it mention because our first round of research has combined the Common Criteria with the SARM to achieve an approach to standards-based security requirements engineering. While SARM endeavors to develop a security architecture, as a first step it allows the development of a set of well defined security requirements. These requirements, applied to the development of security controls at the interface points of domains in a network may result in a security architecture. Lynas and Sherwood both make the point that an architecture is far more than “boxes and wires”. It requires the development of the abstraction of basic business requirements into a specific and well-defined and secured network system. The extent of what we mean by “well-defined” is determined by, obviously, the end-points of the architecting process.

For example, if we terminate our process by translating the requirements developed using the SARM model and matrix into perceived “best practices”, we have defined the solutions to the risks identified in our SARM research in terms of subjective experience. While all solutions not formally derived fit this description to some degree, it would be most effective and less open to criticism if we defined those solutions in terms of some relatively accepted set of standardized criteria.

Such sets of standardized criteria are abundant. Early approaches such as TCSEC offer security guidelines of one sort. Within individual disciplines such as banking, there are sets of security related specifications and standards. If we can tie the SARM approach to these standards we can arrive at a very specific, standards-based set of security requirements. The challenge, however, is to view the network as an entity as opposed to a collection of entities.

The Common Criteria offers us an acceptable option for the testing of an approach that marries the requirements gathering capabilities of the SARM and the need to refine those generalized business and technological requirements into a set of very specific security function requirements. There is, however, no reason why this approach could not be applied to other sets of security standards.

---

## 3.0 Hypothesis

We theorize first, that, using the SARM model, we can define the security requirements for a bounded network. Thereafter, that network may be treated as a discrete entity. For our purposes we will define a “bounded network” as one where we can define clearly a universe of individual domains as well as a specification for their interaction. The universe itself is considered to be a sort of “super domain” comprising the bounded network.

Lynas [LY99] defines a security domain as:

*“... a set of security elements subject to a common security policy defined and enforced by a single security policy authority.”*

He further goes on to give the following criteria for security domains:

- *“Each domain should enforce its own security policy independent of other domains*
- *The boundary of each domain must be explicit*

- *Control must be exerted at the boundary between two domains”*

Our second hypothesis is that we can apply the more general requirements gleaned from the use of the SARM to some set of specific security standards to arrive at a standards-based set of explicit security requirements. If translated into a reference security architecture these requirements may be used as a benchmark for designing a compliant network and testing that network in operation after its construction. We accomplish this by stating the information developed from the SARM matrix in the specific terms of the standard.

The end result is a reference security architecture that ...

- ... results in an explicit set of security requirements that may be stated in terms of controls (including tools, security products and/or configurations) to be placed at every applicable interface in the subject network
- ... is directly traceable to the business, technical and architectural requirements of the subject network, as well as applicable organizational security policies, and
- ... is compliant with a governing standard or standards, allowing objective testing and auditing of the completed network as a unit, not as its individual components.

The last statement, alluding to the network as a whole rather than its individual components is a key differentiator for this approach. We hypothesize that, because the application of the controls resulting from this method focuses upon individual interfaces, the misconfiguration, alteration of hardware or software parameters, or other undefined changes will cause the security of the network as a whole to suffer. By taking the approach outlined in this paper, we believe that the requirement for each and every component of the subject network to be evaluated at the same level as the complete network or higher need not exist. More important, the characterization of a multi-domain, multi-level network becomes easier and more reliable.

Finally, the application of a specific set of standards from which the final set of security requirements is derived yields sufficient definition and granularity to allow objective testing and auditing to ensure security of the network and compliance with the standard(s).

The definition alluded to by Lynas is more stringently explained by Sweezy [SW00] as *policy domains*:

- *“Within each Security Policy Domain, the entities over which the Policy has dominion must be either named, identified or characterized. These are the assets that are of concern within the scope of the Policy Domain.*
- *Given that the entities to be protected are identified, responsibility for that protection must also be assigned. Therefore, the mechanism(s) that implement the security policy must be defined or at least characterized. These mechanisms may, in fact, be manual or procedural.*
- *The Methodology proposal also drives a concept of separation of policy visibility and authority. Rather than attempt to incorporate multiple variations and considerations within the policy statement of a specific policy domain, it is suggested that the policy that controls the activity within a Policy Domain be explicitly separated from the policy that governs the interoperability of the given Policy Domain with other Policy Domains external to its visibility or authority.”*

If we view Sweezy's approach as a refinement of Lynas, we find that we have exactly what we need to view the network as a whole rather than as a set of individualized components. Bear in mind, however, that Sweezy allows (as does Lynas) sub-domains. While we could continue to create child domains down to the level of the individual component, it is unlikely that, with certain important exceptions (such as firewalls), we would gain anything. In fact, we would, likely, defeat our purpose entirely.

---

## 4.0 Objectives and Approach

We have set the following objectives for this research:

- Collect the generalized security requirements for a specific bounded reference network using the SARM matrix
- State the requirements in terms of the Common Criteria
- Create a protection profile for the reference network as a unit using the Common Criteria
- Apply the reference network security requirements to an actual network

We have several actual networks of varying sizes against which we are beginning this process (which we have tested on a theoretical network). Again, we must emphasize that the selection of the Common Criteria as a target standard is one of expedience based upon opportunities for available target networks. The reader should focus on the process and the use of the SARM model as opposed to our selected standard.

### 4.1 APPROACH

Simply put, our approach consists of the following steps:

- Through the use of interviews, document examinations (such as policies) and examination of network maps, collect security requirements in accordance with the SARM matrix. Each and every cell of the matrix must be addressed and some will be addressed several times. These requirements will be stated in the terms of the interviews or document examinations. Typically this process results in at least one narrative requirement to each matrix cell. Additionally, the requirements will map to one or more interfaces within the subject reference network security architecture.
- Translate the narrative requirement captured in the first step into the language of the applicable standard or standards. In the case of the Common Criteria each piece of information must be stated as a *security assumption*, *security policy* or a *security threat*.
- Develop an appropriate response or countermeasure to each requirement as stated in the language of the applicable standard(s). In the case of the Common Criteria, this is stated as a *security objective*. Each assumption, threat or policy must, under the Common Criteria have at least one security objective addressing it.
- Refine the objectives to the extent that they may be used to define, explicitly and specifically, a control or set of controls that must be present at the applicable interface. In the case of the Common Criteria this comprises a complete set of classes which in turn are composed of families which break down into components. The components further break

down into functional elements that the Common Criteria defines as: “...a security functional requirement that if further divided would not yield a meaningful evaluation result.” [CC99] For our purposes we will apply the control(s) described in the security functional element at the appropriate interface(s) as well as at the domain level.

The result, at this point, is a very explicitly defined set of security requirements applied to each interface and domain of the reference model. We can now take the domain-based reference model and apply the requirements to an actual network.

## 4.2 A SIMPLE EXAMPLE

The following example is based upon a theoretical network and uses only a single cell from the SARM matrix for simplicity and to conserve space. See Figure 4-1

	ASSETS	PROCESS	LOCATION
CONTEXTUAL			
CONCEPTUAL			
LOGICAL			Security domain definitions and the security associations between them
PHYSICAL			
COMPONENT			
OPERATIONAL			

Figure 4-1

To illustrate, the Venn diagram of the simple network for which we will define specific security requirements is shown in figure 4-2. The purpose of the external connection is the transfer of files into the trusted domain.

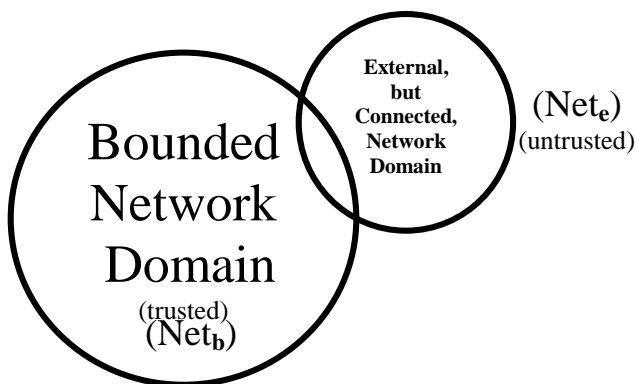


Figure 4-2

Note that, if we are to maintain a bounded network,  $\text{Net}_b \cap \text{Net}_e$  is acceptable (if appropriately defined), while  $\text{Net}_b \cup \text{Net}_e$  is not. Thus, we must define, explicitly and specifically, the security requirements for the interface between domain  $\text{Net}_b$  and domain  $\text{Net}_e$ .

According to the SARM matrix, at the *Logical* layer the *Location* attribute describes the “Security domain definitions and the associations between them”. We will use that cell in our example. It is, of course, possible, that multiple cells (attributes) of the SARM model will apply to a set of security

requirements at a particular interface. For our example, however, we will focus only on the Logical layer/Location attribute.

In the first step of informal data gathering, the requirements might look like this:

1. The internal network domain may connect to an external network domain for the purpose of file transfer.
2. If the external domain is not trusted, only data flow initiated and controlled from the trusted domain will be permitted ( $\text{Net}_b \Rightarrow \text{Net}_e$ )
3. The intersection of the two domains ( $\text{Net}_b \cap \text{Net}_e$ ) must be treated as an isolated domain used solely for the purpose of file transfers into the trusted domain.
4. We have an organizational security policy that defines the process of file transfers into the trusted domain.
5. We use an FTP server on the DMZ of our firewall for this purpose. The DMZ is on a “third leg” of the firewall with its own rule set.

Our next step is to state the narrative requirements in the terms of the applicable standard, in this case the Common Criteria. In the Common Criteria, each requirement must be stated either as a security assumption, a security threat or a security policy. For simplicity we will designate each requirements as A (assumption), T (threat), or P (policy). We will, in the following step, designate objectives as O. There may be, of course, other assumptions, policies and threats associated with this example. However, we will limit them here for simplicity and space considerations.

- **A.Connect:** There will be one or more external connections to the trusted domain for the purpose of file transfer.
- **P.DateFlow:** All data flow between the trusted domain and any untrusted domain must be initiated and controlled from the trusted domain.
- **P.Isolate:** All files transferred into the trusted domain from an untrusted domain must be held in an intermediate isolation area until they can be verified as safe for use in the trusted domain
- **A.Policy:** There is an organizational security policy that defines the process of file transfers into the trusted domain.
- **T.Penetrate:** An attacker can penetrate the FTP server
- **T.RogueCode:** An attacker can place a file containing rogue code, such as viruses or Trojan horses, in the isolation area of the FTP server

For simplicity in assigning objectives to each of the above, we’ll create a small matrix. This will help us avoid skipping an assumption, threat or policy that requires an associated objective. See figure 4-3.

Security Environment	Security Objectives
A.Connect	
P.DataFlow	
P.Isolate	
A.Policy	
T.Penetrate	
T.RogueCode	

Figure 4-3

We now need to develop objectives to satisfy each of the Security Environment requirements. For example:

- **O.Connect:** There will be a method of allowing users in the trusted domain a safe process of file transfer into that domain
- **O.DataFlow:** The firewall between the trusted and the untrusted domain will not permit data flow from the untrusted domain to the trusted domain that is not initiated by a user in the trusted domain.
- **O.Isolate:** There will be a file holding area, isolated from, but under direct control of, users in the trusted domain for the purpose of verifying the safety of all files entering the trusted domain.
- **O.Policy:** The policy, *File Transfer Procedures* defines the process of transferring files from the untrusted domain to the trusted domain.
- **O.Secure:** The firewall is configured and tested to resist penetration and abuse by attackers.

Applying the above objectives to our new matrix we get figure 4-4:

Security Environment	Security Objectives
A.Connect	O.Connect
P.DataFlow	O.DataFlow
P.Isolate	O.Isolate
A.Policy	O.Policy
T.Penetrate	O.Secure
T.RogueCode	O.Secure

Figure 4-4

Finally, we must associate each of our objectives with one or more security functional requirements from the Common Criteria. Note that not all possible security functional requirements are represented. Additionally, certain functionality peculiar to the Common Criteria (such as determination of Evaluation Assurance Level – EAL) are not addressed in this example to maintain the generic value of this approach. If the Common Criteria were, indeed, the selected standard, all of the process would need to be addressed. The same, of course, is true of other standards.

Because the applicable standard (and any processes associated with it) takes over once the generalized security requirements for each attribute of the SARM matrix are determined, the application of the

standard can be undertaken independently of SARM. In any case, the ultimate result is a set of standards-based security requirements, traceable to the organization's stated requirements, applicable to the individual interfaces within the target network and acceptably granular to allow implementation and testing.

Because the individual interfaces interconnect using communications protocols, consideration of communications between devices and domains, as well as the devices themselves yields a network definition that can be evaluated as a single bounded domain. Note that the term "assignment" used below refers to specific requirements, such as explicit policies, to be added during this process.

- **O.Connect:** Common Criteria Component FDP\_ITC.1.3 -The target of evaluation security functions shall enforce the following rules when importing user data controlled under the security function policy from outside the target of evaluation security function scope of control: [assignment: *additional importation control rules*].
- **O.DataFlow:** Common Criteria Component FDP\_IFF.1.5 - The target of evaluation security functions shall explicitly authorize an information flow based on the following rules: [assignment: *rules, based on security attributes, that explicitly authorize information flows*].
- **O.Isolate:** Common Criteria Component FDP\_SDI.2.1 - The target of evaluation security functions shall monitor user data stored within the TSC for [assignment: *integrity errors*] on all objects, based on the following attributes: [assignment: *user data attributes*] **AND** Common Criteria Component FDP\_SDI.2.2 - Upon detection of a data integrity error, the target of evaluation security functions shall [assignment: *action to be taken*].
- **O.Policy:** Common Criteria Component FDP\_IFC.2.1 - The target of evaluation security functions shall enforce the [assignment: *information flow control security function scope of control*] on [assignment: *list of subjects and information*] and all operations that cause that information to flow to and from subjects covered by the security function scope of control.
- **O.Secure:** Common Criteria Component FDP\_ACF.1.4 - The target of evaluation security functions shall explicitly deny access of subjects to objects based on the [assignment: *rules, based on security attributes, that explicitly deny access of subjects to objects*].

This process yields a set of specific security requirements that can be applied to a reference model for the type of network we intend to build. Applying these security requirements to the applicable interfaces in the target network defines the appropriate controls in that network.

---

## 5.0 Conclusions

From our early experience with hypothetical networks using the Common Criteria, we have concluded that this process appears to yield acceptable results. Our objective has been to develop a set of security requirements that, when applied to a bounded network environment, yields a reference model against which a network intended to be compliant with the reference model can be tested.

There is a question of why to use a reference model. Why not apply this approach directly to the "real" network? The answer is that the reference model addresses the intended set of business and technical requirements for the eventual network architecture. That network likely is in place and will become the subject of reengineering to meet the predefined security and operational requirements. Those anticipated requirements are built into the reference model. The planned network architecture may or

may not (probably is not) be consistent with the existing actual network. The intent is to build an idealized model, addressing the organization's business requirements, against which an current network may be compared, reengineered and tested.

---

## 6.0 Future Work

This approach is, clearly, in its infancy. Its youth offers fertile ground for additional experimentation and refinement. For example, while this method has been used in the hypothetical development of Common Criteria Protection Profiles, it has neither been applied to a "real" network nor has it been applied in a standards environment other than the Common Criteria. Therefore, the author sees several areas as inviting for future work.

- **Application to real networks.** The author currently is beginning work on a number of actual networks, ranging in both size and function.
- **Application using standards other than the Common Criteria.** This is a very fertile area for research. The author is exploring opportunities to test this approach with standards in both the financial and healthcare communities.
- **Extension to use with multiple standards.** This is another interesting area since it could encompass defacto standards prevalent in many different disciplines while ensuring compliance with formal standards.
- **Formal proof of the SARM.** This is an area of great interest to the author since formal validation of the SARM model would lead to the ability to characterize formally a set of security requirements that could be "plugged in" to a reference model.

---

## 7.0 Author

Peter Stephenson is the director of technology for the Global Security Practice of Netigy Corporation in San Jose, California. He is the author of several books and numerous articles in computer trade publications. He also is a PhD candidate at Oxford Brookes University, Oxford, UK. Prior to joining Netigy in 1999, Mr. Stephenson operated a security consulting practice for 15 years. He is the developer of the Intrusion Management model for information protection and the VAST process for network vulnerability analysis.

---

## 8.0 References

[SH99] *SABSA Security Architecture Development Methodology*, John Sherwood, et al, published by Enterprise Networking Systems and available from Netigy Corp.

[LY99] *How to Design Security Architecture*, David Lynas, training course developed for Netigy Corp.

[AM83] *Security Kernel Design and Implementation: An Introduction*, Stanley R. Ames Jr., Morrie Gasser, Roger R. Schell, published in "Computer"

[SU97] *Secure Computing*, Rita C. Summers, Chapter 6, pub. McGraw-Hill

[CC99] ISO/IEC 15408-2: 1999 (E), Information technology - Security techniques -- Evaluation criteria for IT security -- Part 2: Security functional requirements, published by the Joint Technical Committee ISO/IEC JTC 1, *Information technology*, in collaboration with Common Criteria Project Sponsoring Organizations.

[ZA87] "A Framework for Information Systems Architecture." John A. Zachman. IBM Systems Journal, vol. 26, no. 3, 1987. IBM Publication G321-5298

[SW00] *Security Topology Modeling Proposal*, a talk given by Michael Sweezy, Lockheed Martin Corporation, DCCIS Symposium, December 2000

Trademark Notice: SARM, SABSA and S-TRAIS are trademarks of Netigy Corporation and are used here with permission.