ABSTRACT
Expressing security controls as functional requirements aids in the verification step of security certification and accreditation. Distributed, multi-component systems or systems of systems (SoSs) are difficult to verify because the security controls must be understood in functional terms with respect to their local effects on components, their global effects on the SoS, and their effect on component information exchange. In this paper, we define a process to formulate functional requirements from security controls with SoSs as the target. The process starts by extracting model elements associated with assets, functions, organization variables, and external influences. These models are composed across a set of security controls and normalized to maintain consistency and remove redundancies. We apply the models to SoSs to provide essential details to their specification in functional requirements. The objective is to reduce ambiguity when verifying SoSs as well as minimize recertification efforts when the system or security expectations changes.

Categories and Subject Descriptors
K.6.4 [System Management]: Management audit

General Terms
Management, Security, Standardization.

Keywords

1. INTRODUCTION
Networks and information systems have grown exponentially into large scale, software intensive systems with multiple, distributed components and services. Security has become a larger issue with the democratization of technology and information. Security controls are one mechanism where organizations choose which accepted techniques to employ to secure their systems. The National Institute of Standards and Technology (NIST) SP 800-53 has established generally accepted security controls for “Federal Information Systems and Organizations” [8]. NIST defines security controls as “…the management, operational and technical safeguards or countermeasures employed within an organizational information system to protect the confidentiality, integrity, and availability of the system and its information” [8, p. 12]. The Department of Defense augments the NIST security controls with information assurance controls in the DODI 8500.2, “Information Assurance Implementation” [3]. Similarly, the DODI 8500.2 defines an information assurance (IA) control as “An objective IA condition of integrity, availability, or confidentiality achieved through the application of specific safeguards or through the regulation of specific activities ... ” [3, p. 20].

Part of the security life cycle includes determining how and to what extent the security controls are satisfied. When multiple, interacting components and services are involved, verifying the system of systems (SoS) that satisfies chosen security controls increases in complexity over standalone systems. This complexity is because the controls must be examined in terms of their different applications to the overall system, the independent components, and their information exchange. Previous research maps the text of the controls to risk components (e.g. assets and criticality) to correlate with control categories [4]. Naldurg et al. [7] model their security controls through the explicit modeling of insecure states. Other research creates a UML meta-model to represent component policies as attributes, behaviors, and mechanisms to instantiate as non-functional security requirements become known [5].

With SoSs, new components may be added, existing components may be upgraded altering internal policy and behavior expectations, connections may be reconfigured to change communication strategies, and new regulations may require altering security needs. If a security control is chosen for a system, then it must be verified in the same manner to assure its consistent application even when the system changes. Brezillon et al. [1] explore contextual-based security research. Their research develops contextual-based models to develop adaptive security policies. Where their method is more granular, this process breaks down the pieces of the system to make the more volatile elements more explicit. When these changes occur, they will not affect the security of the resultant system, as long as the changes are made within given security control parameters.

The goal for this research is to express security and information assurance control into a uniformly, verifiable form so that they can be easily composed to form functional security requirements. Ambiguous requirements limit repeatable verification. One way to combat this ambiguity is the use of formal expressions. Unfortunately, using formal methods introduces complexity and often reduces usability. As a compromise, we have devised a methodology to extract security controls into the model elements of assets, functions, organization variables, and external influences. The extraction highlights key entities for use in verification. Given the set of security controls that deal with a policy issue, we compose and
normalize the model elements to equate semantics and remove redundancies. The resulting entities provide the core predicates of the formal requirements, allowing them to be stated with less ambiguity. We then apply these predicates to SoSs for proper local, global, and interactive parameterization. Finally, we formulate functional requirements related to one or more security controls. Unlike Haley et al. [10], this method creates requirements that focus on implementation for all security and IA controls. The process increases understanding of the differences among the model entities and their use in verification, especially with respect to abstract properties for the early identification of SoS design problems.

2. EXTRACTING CONTROL ENTITIES

In this section, we introduce a security control model to which we initially extract model entities from the controls. These entities are later used to formulate a set of uniform functional requirements for a SoS. However, alone they offer insight into the essential elements that comprise the security control as well as the strength of presence of those elements across multiple controls.

Figure 1 depicts a NIST security control auditing its information systems. The objective of auditing, from the perspective of securing an information system, is to analyze information collected from the system to determine what actions took place and who was responsible. Thus, if the organization believes that auditing certain information is important for its SoS, then it will employ these controls and others that are complementary. Audit trails are typically created and stored to assess if incidents have occurred while ensuring that the system is running at optimal levels. Auditing can be a daunting task for SoS, especially when Web services are used [2], because additional processing to logs and events transform the information needed.

![Image](image.png)

**Figure 1: Audit Security Control AU-6**

We define a Security Control Model, \( \sigma = (A, V, I, F) \) where
- \( A \): the set of assets embodied in the security control.
- \( F \): is a set of functions of the information system and its components embedded in the security controls and applied to the assets and organizational variables as dictated by the influences.
- \( V \): set of organizational variables chosen for the control.
- \( I \): the set of all external influences on the requirements that conform to the current situation expectations and may later change if the situation changes.

We use the definition of assets in [6]: “Data transmitted and stored on the computers and transmitted over the communication lines and the software used to process that data are considered cyber assets.” Functions are derived from activities and capabilities assigned to the information system and/or its components by the security controls. Functions are also assigned by the organization using the security controls. We model organizational variables according to the understanding of the italics sections expressed in the control which is a direct reflection of the choices to be made for that control. Influences are the concerns that define the parameters instantiated in the requirements, such as risk assessment, management expectations, and company regulations. Affected parameters are related to finer-grained information that feeds into the assets and organizational variables such as events, time stamps and user identities. In the remainder of the section, we exemplify the construction of \( \sigma \) for a set of audit security controls.

AU-6 in Figure 1 guides both the organization and the information system (indicated by the added underline). The identified asset for AU-6 is the set of audit records (see first statement of Figure 1), which includes the single record as an asset, so we explicitly specify it.

\[ A_{AU-6} = \{ \text{audit record, ARset} \} \]

The system is responsible for audit review, analysis, and reporting, as well as their integration (see connected boxes). The functions prescribed by AU-6 include

\[ F_{AU-6} = \{ \text{review()}, \text{analyze()}, \text{report()}, \text{integrate(review()), analyze()}, \text{report()} \} \]

Since extraction positions the relevant information in the model, we wait to specify the parameters of the functions until normalization.

The organization variable in AU-6, as seen in italics, is

\[ V_{AU-6} = \{ \text{frequency} \} \]

where \( \text{frequency} \) is the assigned organization time to a designated process \( f \in F \). In AU-6, \( f = \text{review()} \). We associate external influences with reasons for change. In many security controls, risk assessment is the main proponent for change, since it drives the decisions behind using and configuring the security controls for the SoS. Thus,

\[ I_{AU-6} = \{ \text{risk}(F, V, A) \} \]

where all model elements can be affected by risk management.

![Image](image.png)

**Figure 2: Audit Security Control AU-2**

From Figure 2, we extract the following model elements from AU-2, which focuses on the auditable event, with no discernible assets.

\[ F_{AU-2} = \{ \text{audit()} \} \]

\[ V_{AU-2} = \{ E, \text{frequency} \} \]

where \( E \) is a set of auditable events, including the null event, and \( f = \text{audit()} \).

\[ I_{AU-2} = \{ \text{threat}(F, V, A), \text{risk}(F, V, A), \text{mission}(F, V, A) \} \]

In Figure 3, AU-3 focuses on the content of the audit record. The organization also chooses the component that performs the central management of the audit records. The security control model for AU-3 has no external influences explicitly defined.
normalization process. This process first unifies the individual sets. Then it examines each set given its semantic intent to uncover redundancies or inconsistencies. Though we only cover a few audit security controls, the composed model (denoted with the subscript M) holds sufficient information for initial requirements formation. The composition is shown below.

\[ A_M = \{\text{audit record, ARset, audit trail}\} \]

\[ F_M = \{\text{audit(), review(), analyze(), report(), integrate(review(), analyze(), report()), manage(), generate(), compile(), monitor(), create(), alert(), disable()}\} \]

\[ V_M = \{\text{frequency, E, central_mgr, content}, \text{C}, \text{tolerance}(t_1,t_2), \text{configuration}\} \]

\[ I_M = \{\text{risk}(F_M, V_M, A_M), \text{threat}(F_M, V_M, A_M), \text{mission}(F_M, V_M, A_M)\} \]

The combined assets, organizational variables, and external influences are not redundant based on their definitions and expected use. Semantically, the functions audit and generate are the same because they both result in an audit record based on an auditable event. Creating the audit trail and compiling audit records into an audit trail also are redundant. Similarly, report and alert can be grouped into a single function. The monitor function subsumes review and analyze since monitor is continuous. The resulting functions are shown in Table 1.

### 4. SOS APPLICATION

Applying the normalized model to the SoS means that the parameters must be established for the functions based on the model variables. In addition, we have to determine if each function applies locally, globally, or to the interactive information exchange among components. From this perspective, we assign to each component two interactive functions, \(c_1\).send(d,c2) and \(c_1\).receive(c3,d), for \(c_1, c_2, c_3 \in C\) and data, d. (Note that a single component may have multiple interface ports to which distinct send and receive functions would be assigned.) Interactive functions require the designation of two components. The remaining eight functions from the composite model of the five security controls in Section 3 appear in Table 1 with their SoS application designation. The assignment facilitates parameter quantification. Local functions require a single component designation for all components in the system. A global function requires a system designation. Those functions that are local to a component with global information generally mean the existence of a specific component that has access to a global state.

<table>
<thead>
<tr>
<th>Function</th>
<th>Assigned</th>
<th>Rationale</th>
</tr>
</thead>
<tbody>
<tr>
<td>monitor</td>
<td>global</td>
<td>continuous record review and analysis</td>
</tr>
<tr>
<td>alert</td>
<td>global</td>
<td>report on unusual activity across system</td>
</tr>
<tr>
<td>integrate</td>
<td>local, global</td>
<td>local to the integrating component with global information</td>
</tr>
<tr>
<td>manages</td>
<td>local, global</td>
<td>local to the managing component with global information</td>
</tr>
<tr>
<td>generate</td>
<td>local</td>
<td>each component generates its own audit records from designated events</td>
</tr>
<tr>
<td>compile</td>
<td>local, global</td>
<td>local to the central manager that creates the audit trail with global information</td>
</tr>
<tr>
<td>disable</td>
<td>global</td>
<td>disables the entire system when a serious violation occurs</td>
</tr>
</tbody>
</table>

### 3. NORMALIZING MODEL ELEMENTS

Given sets formed by extracting the assets, functions, organizational variables, and external influences for a policy focus or set of security and IA controls, we perform a
5. FORMALIZING CONTROLS

Below we show a sample of the formal requirements that can be expressed given the resulting model information. Assume the colon is equivalent to the statement “is of type”. Let contentAudit_record be a simple ordered pair (event, timestamp) where event : E and timestamp : Time. Let ARset exist at the local component level, to be referred to as c.ARset.

1. Every SoS component can generate an audit record for designated events
   \[ \forall c : C, e : E, a : audit_record | c.generate(e.t) \Rightarrow a.event = e \land a.timestamp = t \land a \in c.ARset \]

2. Every SoS component generates audit records at a designated frequency
   \[ \forall c : C, a1 : audit_record | \#s.ARset \leq 1 \lor (\exists a2 : audit_record)[a1, a2] \subseteq c.ARset \Rightarrow a2.timestamp - a1.timestamp \leq frequency_c.generate(a) \]

3. Every SoS component can send their audit records to the central manager who can receive them
   \[ \forall c : C | c.send(c.ARset, central_mgr) \lor \forall c : C | central_mgr.receive(c, c.ARset) \]

4. The central manager component compiles the SoS component audit records into a system-wide audit trail
   \[ \forall c : C | |c.ARset| = s.ARset \land central_mgr.compile(s.ARset) = s.audit_trail \]

5. The audit records in the system-wide audit trail have timestamps within defined tolerances
   \[ #s.audit_trail \leq 1 \lor (\forall a1: audit_record, \exists a2: audit_record, t1 : Time, t2 : Time | tolerance_{t1,t2} \land t1 \leq a1.timestamp \leq a2.timestamp \leq t2) \]

6. ACCOMMODATING CHANGE

Because risk, threat, and mission assessment occur at strategic intervals for a system and organization, an assessment outcome can trigger less acceptance of the current risk profile of the system. Manifestations of the direct influence of the change appear in the organizational variables. These changes impact the requirements differently. If the frequency of the event audit changes, that impacts every component because they must log an event at a particular time. Similarly, the central manager must ensure the audit trail remains properly time-correlated. In contrast, if the frequency of audit trail review changes, then it impacts only the central manager’s requirements. If the events to log (E) or the internal structure of the audit record changes, then all components are affected, but in different ways. The components must ensure that their record structure is up to date and that they are collecting the right events given current risk assessments. The central manager must update its monitor and alert processes, but integration of these functions may not be reproduced and distributed. Therefore, by developing the requirements with knowledge of how changes due to external influences affect assets, functions, and organizational variables, the impact of the change can be directly targeted, reducing re-verification efforts.

7. DISCUSSION AND CONCLUSION

In this paper, we bridge the functional requirements gap between what the organization must require of the system, what the information system must do, and what each component in the system must do with respect to security controls. We introduce a strategy for extracting essential model entities, composing and normalizing them across a set of complementary security controls, associating and parameterizing the resulting functions according to their reference in a SoS, and formulating requirements that reflect the initial set of controls. The objective is to form a set of encompassing functional requirements that govern a SoS at the local, global, and interaction levels. The move to more formal requirements can reduce any ambiguity in determining if the SoS satisfies the security controls. At the same time, it reduces the need to reexamine all requirements when external influences force assets, functions or organizational variables to change. Instead, only those requirements affected by the change are targeted for re-verification.

8. ACKNOWLEDGEMENTS

This material is based on research sponsored by the Air Force Office of Scientific Research, under agreement number FA-9550-09-1-0409. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views expressed in the paper are those of the authors and do not reflect the official policy or position of the Department of Defense or U.S. Government.

9. REFERENCES


