Assessing and Managing Quality of Information Assurance[1]

Partha Pal
BBN Technologies
10 Moulton Street
Cambridge, MA 02138
ppal@bbn.com

Patrick Hurley
Air Force Research Laboratory
525 Brooks Road
Rome, NY 13441-4505
patrick.hurley@rl.af.mil

ABSTRACT

The connection of coalition systems has many challenges, one of the most important and the one being address by this paper is the lack of understanding of information assurance (IA) in a coalition environment. This paper presents an approach to managing this coalition IA risk using a taxonomy to organize readily available observations and measurements that are potential indicators of a system's level of information assurance (IA). This paper also describes how this taxonomy can be used for runtime mission-oriented assessment and management of IA assets. In this context, a mission refers to a specific set of tasks carried out using the system by a group of users cooperating towards achieving a common objective, and IA refers to the users' level of confidence that the system can be entrusted with their respective tasks.

1.0 INTRODUCTION

Modern information systems routinely incorporate security mechanisms such as firewalls, antivirus scanning tools and mechanisms for user authentication and authorization. Advanced mission-critical systems often incorporate security mechanisms aimed to maintain mission-specific security requirements (expressed in terms of availability, integrity and confidentiality) organized in a coherent manner that facilitates defense in depth. However, quantifying the contribution of the included security mechanisms or the underlying survivability architecture in improving the system's ability to thwart, defend or survive cyber attacks has eluded researchers and practitioners for a long time. As a result, information assurance (IA)—assurance that the security mechanisms are effective, and the system can be entrusted with critical information processing tasks—is largely qualitative, and is determined primarily by offline evaluation processes such as analyses (e.g., NSA INFOSEC Assurance Capability Model or CERT/CC Security Capability Model), testing (e.g., penetration testing, fuzzing or fault injection), modeling (e.g., modeling the system behavior and model-based studies to determine attributes like mean time to failure based on certain attack profiles) and experimentation (e.g., red team experiments) that are detached from actual deployment and operational missions. Consequently, at runtime or during mission execution, arguably the time when it is most critical to be assured about the system, IA takes on an all-or-nothing flavor (i.e., either the entire system is assured or it is not) and is largely dependent on the user's perception (as opposed to any real and objective measure). The risks posed by attack-induced failures and security-compromises, environmental threats such as the release of a new virus or discovery of a new vulnerability, and user-made decisions to change security settings or to bypass security entirely are neither well understood nor considered in such perception-based assessments.

Military systems, including the information systems NATO relies on for its missions, are constantly under the threat of malicious attacks, and therefore need to incorporate an appropriate collection of defense mechanisms for their security. At the same time, there has been an increasing realization that mission

[1] This work is funded by AFRL under the Contract No. FA 8750-08-C-0196. The views expressed are those of the authors and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

stakeholders need to know (estimate) the level of assurance accorded by the defense, especially on a continual basis (during mission operation). Without such knowledge and given the *perception-based* and *all-or-nothing* view of IA, war fighters and system owners might:

- Incorrectly decide not to use the information system, because they fear a compromise,
- Change the system configuration improperly without understanding the impact on the mission,
- Continue to use the system without knowing the extent or significance of a compromise,
- Think the system is protected (and not act on reported events) when it is not, and
- Cannot effectively execute their tasks in a highly dynamic environment where coalitions and joint missions are quickly formed; mission objectives, information system configuration and roles of mission participants can change during a mission.

All of these cases incur risks. With the increasing dependence on network centric information systems, addressing these risks has become an urgent requirement. The work being performed by researchers at BBN and AFRL to develop a framework to assess and manage the quality of IA offers a potential solution.

The high level goal of this research is to show that existing survivability architecture and the constituent defense mechanisms, along with the sensing and reporting infrastructure of existing system management tools, can be leveraged to provide a meaningful and continuous mission-oriented assessment (CMA) of assurance. The assessment is focused on estimating the level of assurance provided by the system at any given point against a desired level—i.e., the quality of IA as opposed to an absolute quantification of IA. More specifically, we would like to demonstrate that information systems can be instrumented with suitably placed probes and aggregating mechanisms such that the aggregating mechanisms are able to continuously indicate whether the system is operating at a required level of assurance based on measurements and observations reported by the probes. The required level can vary over time during a mission and can be different for different stakeholders. An additional goal is to support adaptive behavior leveraging the newly developed continuous assessment. Continuous assessment combined with adaptive behavior will demonstrate a new, dynamic way to manage IA and enable interoperation with existing QoS mechanisms so that where appropriate, assurance and service delivery can be traded off dynamically (e.g., sacrificing encryption for faster response). We also envision a specific “assurance engineer” role to be played by security specialists, responsible for capturing assurance requirements, identifying the measurements and observations necessary for continuous assessment, and instrumenting the system with appropriate probes and aggregator nodes that collect and process the observed/measured information according to the assessment scheme devised for the mission.

This paper reports research that is very much work in progress. The main contributions of this research report are:

- A taxonomy and organization of factors that contribute to mission-oriented assessment of IA
- A methodology to perform the assessment
- An initial proof of concept for mission-oriented continuous assessment capabilities, and
- An early glimpse into a framework for assurance/service delivery tradeoffs.

We argue that the technology developed by this work fills an important gap that currently exists in mission critical information systems. The continuous mission-oriented assessment and the accompanying adaptation capabilities will make information systems of the future more effective for dynamic missions and facilitate dynamic adjustment of system and user behavior appropriately when the delivered level of IA falls below the required level. Mission stakeholders will also have a realistic and objective way to determine in real-time whether they can trust the system with their mission-critical task and the impact of turning off or tuning security configurations on their missions.

2.0 A FRAMEWORK FOR CONTINUOUS MISSION-ORIENTED ASSESSMENT

Past research has not been able to develop a usable quantification of system security and assurance. State of practice in evaluation of security and information assurance is somewhat *subjective* and *qualitative.* In the most advanced form, it is a combination of various *offline* evaluation processes including,

a) Analysis of the system against prescribed guidelines such as the NSA INFOSEC Assurance Capability Model, CERT/CC Security Capability Model, B-Secure Security Maturity Model or NIST SP 800: Security Self Assessment Guide for IT Systems;
b) Various types of testing including penetration testing, fuzzing and fault injection;
c) Modeling of systems and system behaviour and model-based studies to determine attributes like mean time to failure based on certain attack profiles, and
d) Experiment based approaches such as red team exercises.

Furthermore, it is generally accepted that security can not be absolute, only adequate but the definition of *adequate* depends on context, and may change as mission progresses, which makes measuring security even more complicated. Finally, modern systems are complex, with too many moving parts, are distributed, and have multiple stakeholders that may have different security interests.

A single, universally acceptable quantification of system security and assurance is therefore unattainable. We need to break the assessment problem down to manageable levels (divide and conquer).

2.1 Multi-Dimensional Assessment Space

We claim that the assessment space is multi-dimensional, and usable assessment of IA must be *continuous*— i.e., assessment must be done in an ongoing manner while the mission is executing, and *mission aware* — i.e., assessment must take into account that there are multiple stakeholders in the system, whose IA interests may vary over time during the mission.

In defining the various dimensions of the multi-dimensional assurance space, two obvious dimensions are therefore *time* and *stakeholders*. Changes in a mission's assurance requirements can be based on elapsed time or mission events. In our framework, we support variation of IA requirements based on both elapsed time as well as mission events. In terms of stakeholders, our framework currently supports three representative classes of stakeholders: *commander* (with an ownership stake), *warfighter* (end-user stake) and *operator* (system-administration stakes), which represent the 3 possible values in the stakeholder dimension. Not every stakeholder is interested in the entire system—typically a stakeholder is interested in an end-to-end capability or a subsystem (a subset of physical components and networks) and/or a subset of services offered by the system. Therefore, *spatial scope*, capturing the hosts, networks and applications/services that are of interest to a stakeholder is the third dimension of our assurance space. Classically, security of a system is described in terms of *confidentiality* (C), *integrity* (I), and *availability* (A) of the services it provides and/or the information it handles [1]. Availability and confidentiality is defined in terms of *authorized* users, but does not consider the strength of access control and authorization mechanism involved i.e., whether it was open access that authorizes any requester (which incidentally will conflict with any confidentiality requirement), or authorization was based on a user-provided password or validating a common access card (CAC) (CAC authentication being stronger than password-based authentication). In our exploration we included *authentication/access control level* (A/A) (we treat authentication and access control together, because access rights are usually authorized based on some notion of an authenticated identity) as another security attribute that is operationally relevant but is not covered, rather assumed by C, I and A. C, I, A and A/A are the last four dimensions (collectively referred to as the assurance dimensions) of our multi-dimensional assurance space.

Mission awareness in our approach comes from the following aspects. First, our assessment determines

whether the quality or level of IA delivered by the system satisfies what is *required* at any given time within the mission, as opposed anything absolute. Second, IA requirements reflect how different mission stakeholders' assurance needs change over time during the mission. Finally, the IA requirements are specified in terms of individual IA attributes over a spatial scope, which provides an adequate level of flexibility and granularity to capture many dynamic mission situations.

Our framework treats IA assessment in a relativistic and quantized manner. The IA requirements are expressed in terms of ordered *levels* such as high, medium or low, where the number of levels depends on the distinct levels the stakeholder needs to describe his requirements. A beneficial side-effect of our approach is the fact it forces the stakeholders to be explicit in terms of their assurance requirements (i.e., what spatial scope, which IA attribute and how it changes over time).

Within this structure, we next describe what measurements and observations can be used to estimate the delivered level of IA, where the delivered level, once again, is an assessment at a specific time in the mission for a specific spatial scope and IA attribute.

2.2 Metric Classes

It is impossible to specify a list of specific measurements and observations that could be universally applicable and available in every mission context. Instead, we focus on general classes of measurements and observations that are useful to determine delivered levels of IA. We claim that modern systems already collect a lot of information, and the minimal pieces needed to determine delivered levels of IA are either already available or can be collected without much difficulty. In our framework, system condition objects represent collected measurements and observations. There are designated nodes in the system that aggregate the values presented by the system condition objects, and use the aggregated values for assessment.

The two mandatory classes of observations and measurements in our framework are called DEF STAT and RES STAT, representing the state of defense mechanisms and resource status respectively.

The state of the defense mechanisms included in the system, whether they are functioning at their intended configuration or have been disabled or degraded, is a primary contributor to the continuous assessment process. If the system is intended to offer a level of security, by design the system must have included appropriate defense mechanisms that are supposed to operate at a designated configuration. The system conditions in the DEF STAT class are meant to observe and report the configuration state of defense mechanisms in the system. Ideally, turning off or changes in defense mechanism configuration resulting from operator action or attack activity should be reflected in the value reported by these system conditions. The key challenge to realize this ideal condition is to make the observing and reporting mechanism sufficiently independent of the defense mechanism such that controlling the defense mechanism does not imply controlling the monitor. Enforcement of OS and network-level isolation policies, stronger process authentication and digital signatures can be employed to ensure that the adversary cannot easily feed fake or incorrect observations.

The state of system resources directly impacts the availability requirements, and since defense mechanisms that provide other aspects of security such as confidentiality and integrity, the ability to meet these requirements are also indirectly affected. Therefore, RES STAT system conditions, focusing on the status of computing resources, specifically CPU, memory and the network, are important for continuous assessment. Modern hosts and network equipments (e.g., routers) already monitor detailed health statistics. System management tools (e.g., open source Nagios [8]) and protocols (SNMP) that can collect and distribute them efficiently are widely available. SNMP version 3 offer stronger security feature such as message integrity, authentication between agent and server, and encryption. The RES STAT system conditions take advantage

of these existing capabilities.

Apart from DEF STAT and RES STAT, our framework accommodates 5 other classes of measurements and observations namely, DEF REP, DEF EFF, EXT, POM and AIQ.

The DEF REP class contains measurements and observations derived from defense mechanism reports. The reports convey information that is distinct from the off/on or configuration state of the defense mechanisms reported by the DEF STAT system conditions. The DEF REP system conditions capture information about unexpected or suspicious incidents and known attack indicators (usually in the form of alerts or logs) that are of interest for continuous assessment. Survivability architectures or security management tools (e.g., open source OSSEC [9]) already provide a way to interface with host based security mechanisms in a secure way, parsing their raw reports and presenting a processed report via a server. Implementation of DEF REP system conditions takes advantage of existing mechanisms such as OSSEC.

The DEF EFF class focuses on effectiveness of defense mechanisms (DEF EFF). Modern systems increasingly combine elements of protection, detection and adaptation [2] in their survivability architecture. In such a system, defense mechanisms engage in actions to thwart, gracefully degrade or recover from the attack in an effort to continue operating (i.e., survive). Under an attack, assessment of the level of assurance must consider whether such responses are effective or not. In our prior work [3] we have demonstrated a mechanism to determine the effectiveness of defensive responses mounted by a survivable system, which provides a starting point for the DEF EFF system conditions.

The external (EXT) class represents the relevant events that happen in the environment in which the system operates and can impact the system's security. Examples include vendor-issued advisories (e.g., Symantec), changes in DEFCON status or in national threat levels (e.g., DHS threat level in the US, or a unified infosec threat level covering NATO member countries) and alerts from security organizations (e.g., CERT) that may imply increased risk to DoD information systems.

The Process and Organizational Maturity (POM) class focuses on the maturity of the software and security engineering process, and the cultural and operational practices of the organization. Our survey of best practices showed that quite a large number of security evaluation methodologies such as NSA INFOSEC Assurance Capability Model (2004), CERT/CC Security Capability Model (2005), B-Secure Security Maturity Model (2003), NIST SP 800 (Security Self Assessment Guide for IT Systems (2001) consider process and organizational factors. This is the motivation to include this class in our framework.

The AIQ class considers Architecture and Infrastructure Quality - how the system is constructed, especially if there are built-in security or defense mechanisms present in the system to protect, detect or manage breaches to basic security properties such as confidentiality, integrity, availability, and authentication/access control. We have shown in previous work [4] that the resiliency of a system against malicious attacks is strongly related to the system's survivability architecture. Some checklists/analyses methodologies do include architectural quality. Common Weakness Enumeration (CWE) is a recent example that highlights the connection between software quality and security assurance. Therefore AIQ is an important category to consider in the assessment of assurance levels.

The next section explains how measurements and observed events can be used to determine the delivered levels of IA.

2.3 Mapping Measurements and Observations to IA Levels

Once again, the function that maps the measurements and observations to the delivered levels of IA is

context dependent. For example, how many levels are applicable for a particular assessment (for a stakeholder for the specific IA attribute with respect to the spatial scope he is interested in) depends on the mission. What classes of observation and measurements are available also depends on the system at hand. Our framework provides a concrete way to use the mandatory metrics DEF STAT and RES STAT within a given system context where DEF STAT determines a *baseline* level of delivered IA and RES STAT modifies the baseline with a *variance*. For brevity, we describe our approach in terms of an example.

Let us assume that for a particular assurance attribute A in the context of a specific spatial scope E of a specific stakeholder H, there are k relevant defense mechanisms. Determination of the k defense mechanisms in such a context is a routine system analysis procedure that security engineers today perform regularly. Let us also assume that H requires 3 levels of A on E during the mission. This is what our approach makes explicit—stakeholders must specify what their IA requirements are.

Some of the k defense mechanisms are distributed in nature i.e., with components on multiple places, and therefore the end-to-end configuration of them will depend on observations and measurements made at different parts of the system. Therefore, we will have n DEF STAT system conditions where n>= k. Each system condition S_i can have values in the range [1, x_i], where the higher values imply stronger security, representing x_i different states or configurations it can operate. The simplest defense mechanisms operate in a binary mode—they are either OFF or ON.

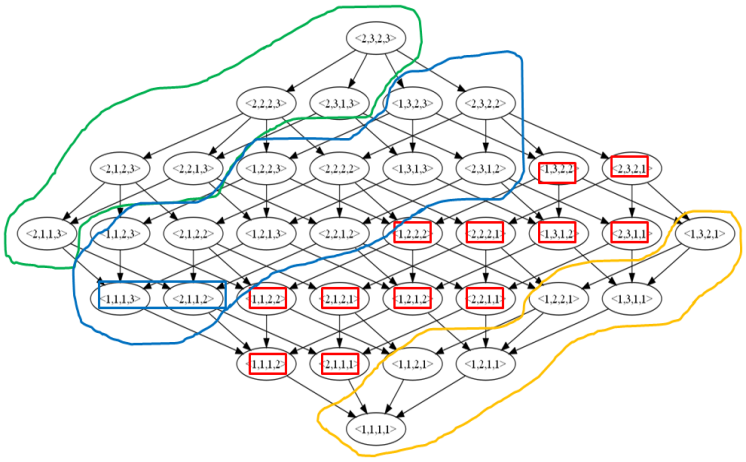

Figure 1: An example DEF STAT state space

With this information, it is possible to construct a graph (Hasse diagram) showing the partial ordering of various possible values of the n system conditions as nodes. Figure 1 shows an example with 4 system conditions s1, s2, s3 and s4, where s1 and s3 are binary valued (they can be either 1 or 2), and s2 and s4 have values in the range [1, 3].

The implicit relationship between the system conditions that reflect observations at different end points of individual defense mechanisms (offering end-to-end capability, such as encryption) imply that a subset of these nodes is *invalid*. For example, if the spatial scope is a service that H is using, and we observe that his side (i.e., the service consumer) of the interaction is set up to encrypt and the remote (i.e., the service provider side) is set up to handle clear text, this configuration is not workable. Invalid configurations are shown by the red boxes in Figure 5, indicating configurations where (s1 reads 1 and s4 reads 2) or (s1reads 2 and s1reads 3) are incompatible. It may also be possible that some configurations are deemed to provide the same level of security. This is shown in Figure 5 by the blue box, indicating that the configurations where (s1 reads 1, s2 reads 1, s3 reads 1, s3 reads 3) and (s1 reads 1, s2 reads 1, s3 reads 1 and s4 reads 2) are *equivalent*.

Given this, it is possible to define a function *f* as a grouping of nodes. The colored bands in Figure 1 show one possible grouping. The green band includes all configurations where s1 and s4 report their highest possible value—in this example, the assurance engineer deemed these configurations providing the strongest

A for E. The configurations in the blue band where either s1 or s3 has their highest possible value were deemed to offer the next level of A for E. The remainder of the configurations is deemed to offer the lowest acceptable level of A for E.

The algorithm to assess the baseline level (which will also be the assessed level in the absence of any variance) then becomes a membership checking: which band contains the current configuration indicated by the observed DEF STAT values? But explicit membership checking on a large graph can be time consuming (the total number of nodes in the graph is the product of x_i s, where x_i is the number of values each system condition can take can be extremely large). To address that issue, we make use of the atLeastAsSecureAs relation, which captures the partial ordering implicit in the DEF STAT space, between two nodes defined as follows:

```
atLeastAsSecureAs([],[]).
atLeastAsSecureAs([F1|R1],[F2|R2]):-
      F1 >= F2,
      atLeastAsSecureAs(R1,R2).
```

In the above code fragment, a node N is defined as a list of values (e.g., N = [a, b, c, d] and the [F | R] notation implies F is the first element in the list and r the remainder of the list (e.g, F= a and R = [b, c, d] for the list [a, b, c, d]). It is straightforward to define the various colored bands in terms of the invalid, equivalent and atLeastAsSecureAs relations.

The variance modification may not be applicable to all assessments. More specifically, the mandatory RES STAT information modifies the delivered level of Availability only. Continuing with the example baseline depicted in Figure 1, it is possible to define availability assessment as a mapping from the system condition values to the 3 levels required by the stakeholder as shown in a tabular form in Figure 2. The 1st column shows the baseline assessment based on the observed state of the DEF STAT system conditions. Columns 2 and 3 show two RES STAT system condition values, one represents the load of the network between the service user and the service provider (Network load) and the other represents the load at the server (Host load). The definition of load is deliberately kept unspecified—it is up to the assurance engineer to use a quantity that is indicative of the network or system load. Existing system management mechanisms routinely provide a composite load indicator for the network, host platform and for application services; the assessment mechanism can accept any of these values. Column 4 shows the variance, and Column 5 shows the overall assessed availability value for stakeholder H for the spatial scope E.

f(.)	Network load	Host load	g(.)	Assessed L (f(.), g(.))
Green	< 60%	< 60%	1	High
Green	>= 60%	>= 60%	0	Medium
Blue	< 60%	< 60%	1	Medium
Blue	>= 60%	>= 60%	0	Low
Yellow	>= 60%	>= 60%	0	Low

Figure 2: Example of variance influencing the baseline

If DEF REP, DEF EFF, AIQ, POM and EXT system conditions are available in a given context, AIQ and POM measurements can be used in determining the baseline as well as the variance. This is because, a robust architecture (e.g., multiple layers of defense) and seasoned users (e.g., who will not click open an arbitrary attachment) offers a higher level of assurance to begin with. Therefore, AIQ and POM measurements have a constant additive impact on the baseline assessment. At the same time, during the mission, the robust architecture and prudent user actions tend to dampen the progress of attack effects. Therefore, these measurements should have a damping effect on the variance function. On the other hand, DEF REP, DEF EFF and EXT measurements and observations only contribute to the variance function—but unlike the AIQ

and POM, depending on their values they may either dampen or reinforce the variance. For example, if we continue to observe high severity reports from defense mechanisms, the variance function should force the assessed value below the baseline. On the other hand, if DEF EFF values indicate that the defense is being effective, the variance function should force the assessed value upwards (towards or above) the original baseline. Similarly, if the EXT observation reports a new exploit spreading in the network, the variance function should push the assessed value downward. However, if the EXT observation indicates that a vulnerability is being patched, the variance function should force and upward revision of the assessed value. In all cases, the extent of the influence i.e., the additive constants or factors need to be determined by assurance assessment engineers, customized for the given context.

3.0 INTEGRATING ADAPTIVE BEHAVIOR WITH ASSESSMENT: TRADEOFFS

Deviation in the delivered level of IA can be caused by attacker activity as well as user activity. Sometimes a user may choose to sacrifice the IA. For example if an F16 of a NATO country is passing by an US AWACS, there may not be enough time for key exchange. The F16 may choose to send data in clear text, or the AWACS may accept an encrypted data drop in an isolated port that it can later use to determine the authenticity of the data obtained from the F16. This involves the mission users consciously changing configuration settings or turning off certain defense mechanisms. Users may affect such changes inadvertently as well. It is desirable to know how such actions impact the level of IA delivered by the system and whether the delivered level still meets the required level. It is also desirable that such deviations from the required level are mitigated promptly. Mitigation usually takes the form of adaptive behaviour—either the stakeholder accepts the degraded level of assurance and changes his behaviour (based on mission role), or the system is reconfigured in some way to get back to the desired level. In either case, mitigating actions are likely to impact quality of service (QoS) in the system. This is the reason we would like our framework to support IA-QoS tradeoffs. This section presents a snapshot of our work so far on this topic.

The QoS space is also multidimensional just like the IA space. Different mission stakeholders may have different QoS requirements, and the requirements may vary during the mission. QoS requirements are expressed in terms of a number of attributes, and the attributes are all about specific services, which is analogous to the spatial scope in the IA space. There is some overlap between the IA and QoS attributes. For example Availability is an attribute in both spaces. The QoS attribute Fidelity is related to the IA attribute Integrity, and the IA interpretation of the attribute Availability includes some aspect of the QoS attribute Timeliness (a delayed response is as good as the service being unavailable).

The measurements and observations that are relevant for QoS can also mirror the structure of the IA metric classes. In fact, measurements and observations from POM, AIQ, RES STAT and EXT can be used in QoS management directly. SERV STAT, SERV REP and SERV EFF can be the QoS analogues of the DEF STAT, DEF REP and DEF EFF IA metric classes respectively. One noteworthy difference is that DEF EFF can be a single system-wide measurement (i.e., how effective the defense has been overall, as opposed

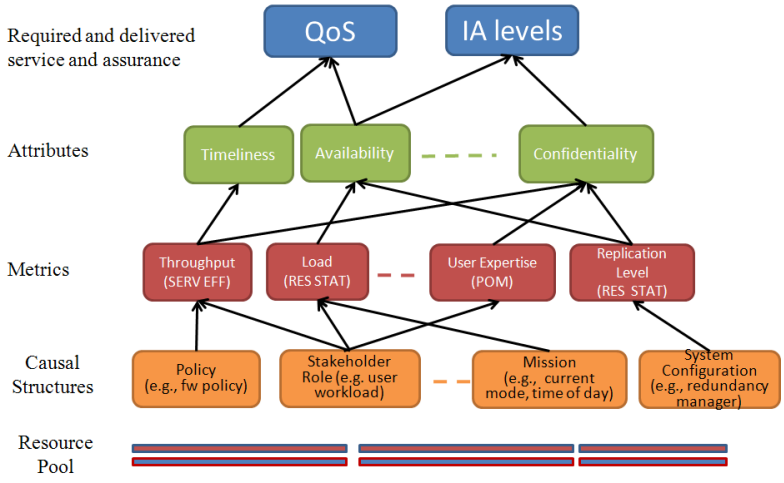

Figure 3: Structure of QoS and IA relationship

to measuring the effectiveness of individual defense mechanisms), but SERV EFF measurements, indicating how efficient service has been, is almost always specific to an individual service.

Prior work in QoS [5] in distributed systems has developed a number of mappings from QoS metrics to delivered levels of service or service quality, mostly involving the Availability and Timeliness attributes and RES STAT and EXT metrics. These mappings are used in QoS-managed adaptive behaviour such as adjusting resource allocation to meet the timeliness requirements despite naturally occurring variations in the resource pool, system load or deliberate changes in mission mode (e.g., a UAV switching from pursuit to battle damage assessment will change what type of images need to be captured and sent and how fast). Impact of such changes on IA has generally not been considered—in most cases, a fixed resource budget e.g., 10% of CPU for security related functions, or 100ms for verifying encryption, is used for resource allocation calculation. Similarly, cyber defensive often do not consider the impact the defense mechanisms may have on service delivery. The very presence of defense mechanisms impacts service delivery because defense mechanisms use the same resources that are needed for service delivery and security functions (e.g., cryptographic operations) are time consuming. A very common autonomic defensive response is shutting down a port or service, which may directly impact service availability and hence the mission. Even with redundancy and failover, adaptive responses such as switching to new replicas, starting or moving replicas can cause a level of service unavailability.

QoS management in complete isolation of IA or vice-versa is impractical. In order to maximize mission success, adaptive behaviour must consider the tradeoffs between IA and service delivery. In our prior work we have demonstrated how advanced middleware can be used for integrating and interfacing with resource management [6] and defense mechanisms [7] to support adaptive behaviour. We are leveraging this work for the plumbing required for IA and service delivery tradeoffs. Formalization of the relationship between IA and QoS, and IA-service delivery tradeoffs are similarly benefiting from the existence of the IA framework described in Section 2 and informed by the prior QoS experience.

Our current view of the relationship between IA and service delivery is captured in Figure 3. Assurance level and service quality, as required or as delivered can be expressed in terms of attributes. As mentioned earlier, some of the attributes are used to describe both IA and QoS. The current set of attributes we are considering are showed in Figure 4. Metrics, i.e., the actual measurements and observations that we use to assess the level of IA or QoS are organized in a number of classes—some of these and specific instances of actual measurements and observations are shown in Figure 3. Note that an individual measurement can relate to multiple attributes. We posit that causal structures, which are often dependent on the specifics of the information system at hand and its operating environment, underlie the metrics and interconnected aspects of service delivery and information assurance. Each of these causal structures may involve a collection of distributed resources and impact one or more measurements and observations.

Attribute	QoS	IA Level
Availability (A)	X	X
Agility (G)	X	X
Timeliness(T)	X	
Fidelity(F)	X	
Confidentiality (C)		X
Integrity (I)		X
Authentication/Access control level (A/A)		X

Figure 4: QoS and IA attributes considered

The causal structures determine what can be observed and measured in a given system. Causal structures can be of various types ranging from applicable policies, roles played by stakeholders, mission objectives and system configurations. Figure 3 shows specific examples of these classes. Some of these causal structures may be clearly visible whereas others may be hidden. However, manipulation of these causal

structures, even at a gross level, is essential for QoS and IA tradeoffs. It is clear that "resources" play a crucial role in QoS and IA, therefore system reconfiguration and resource management actions are two key avenues for effecting tradeoffs. It should also be noted in this context that sometimes the causal structures may expose an interface that offers both measurement and control. A resource management mechanism such as a replication manager may allow a reading of the replication level as well as controlling it.

A key aspect of trading off service delivery with IA is determining when to initiate the tradeoffs and exactly what actions to take i.e., the policies that govern the tradeoffs. In our framework, tradeoffs are triggered only when the system fails to deliver both the required level of IA and the required QoS for a specific stakeholder at a given time for over an intersecting spatial scope. The intersecting spatial scope is important, since that signifies the point of contention—e.g., an end-to-end function whose security and performance both are of importance to a war fighter. Our framework uses simple "preference" rules to guide what should be traded off against what, once a contention is detected. For instance, a war fighter may simply have a rule stating that in a certain mission mode, he always *prefers* QoS over IA.

We are currently in the process of fleshing out the design and developing a proof of concept prototype that will demonstrate IA-service delivery tradeoffs based on contention in the spatial scope and guided by stakeholder specified preference rules.

4.0 RELATED WORK

Most commonly used information assurance assessment techniques evaluate the system offline, when it is not serving a mission. Such techniques make use of published vulnerability reports (e.g., SCAP [12], or security policies and guidelines (e.g., EAL levels based on common criteria [15]).

There are two major aspects to vulnerability based assessment—vulnerability reports and scoring schemes. Vulnerability reports may come from many sources including software or hardware vendors like Cisco and Microsoft, government supported organizations like CERT, or security solution vendors like Symantec. Vulnerability reports from disparate sources are organized and cross-referenced by efforts like the Common Vulnerabilities and Exposures (CVE) [10] in an ongoing basis. For scoring the severity of vulnerabilities, the Common Vulnerability Scoring System (CVSS) provides an open and extensible standard that uses temporal or environmental factors [11]. Since the release of CVSS v2 in the middle of 2007 there has been a trend towards standardization as multiple companies have started to use it, including Symantec, Akamai, Cisco, Qualys, and McAfee. One aspect of the CVSS that caused it to rise above others, other than the fact that it's an open standard, is its ability to be customized for a given system. Security Content Automation Protocol (SCAP) [12] is the current high-water mark for assessing policy compliance and measuring systems and vulnerabilities. While it does not generate the configurations or consistent measurements, it does provide a framework in which they can be evaluated. Using its six constituent standards, which include CVE and CVSS, SCAP takes in descriptions of how the system should be configured and scores them on compliance as well as scores vulnerabilities based on their impact. While vulnerability based assessment can, in principle be done continuously when the system is in operation, as reports are published, and customized for a mission, it is typically done by a human specialist, one vulnerability at a time and scoring a specific part or subsystem at a time. In our framework, vulnerability reports are used as one of the any factors influencing the systems assurance.

Several NIST publications (e.g., NIST Special Publication-800-55[13], Federal Information Processing Standard (FIPS) 140 [14]), the orange book and its successor common criteria [15] offer examples of models and rules that are used to assess the security level of information systems or system components. Common Weakness Enumeration (CWE) is a list of common software weakness that came out of the need to formally

categorize security weaknesses [16], and can also be used in a check-list oriented assessment. Once again, the assessment processes need heavy involvement of human experts, and are typically not done at runtime and does not have a mission focus.

Dependability centric measurements that require long term observation of the system or model based studies to compute metrics like mean time to attack (MTTA), mean time to failure (MTTF), or mean time to recovery (MTTR) offer another way of rating a system. When this type of assessment may involve runtime observation and measurement, it is of little help to decide whether the system can be trusted at runtime and possibly under attack.

Red teaming [17] is another practical approach that is often used to assess the quality of defense. While this is very useful to identify the flaws of the system, rating a system based on the outcome of a red team evaluation can be misleading because the outcome of the red team exercise depends on the capability, motivation and resources of the red team.

Quality of protection (QoP) is another approach where *protection* is used as a quantitative measurement across an entire system. While it provides an umbrella for a number different security related properties (e.g., Foley et al [18] proposed a framework on which to map multilevel security to QoP, Aime et al [19] takes into account vulnerabilities and best-practices to model quality of protection) the ratings are still static.

5.0 CONCLUSION

Many "top ten" lists of cyber-security research problems have included the lack of adequate means and metrics to evaluate and assess security of information systems. A recent one from DHS [20] posits that if we cannot measure, we cannot manage cyber security. Cyber attacks tend to consume and corrupt the very resources that the system needs to function. Attacks often disable or change the configuration of defense mechanisms as well. Despite a lot of information being collected by security and system management mechanisms in the system, mission stakeholders often remain in the dark about the security state of the system and whether or how their mission objectives are impacted.

We provide a methodology and framework to remedy the situation. In addition to offering a continuous assessment of IA, the framework also offers a foundation for IA-service delivery tradeoffs. We expect to demonstrate the feasibility and benefits of continuous mission-oriented assessment and IA-service delivery tradeoffs using prototypes of the various concepts and capabilities described in this paper in the context of a realistic mission scenario. However, significant research and engineering work will still be needed before the proof of concept technology becomes ready for operational use. Scalability in terms of number of hosts and users participating in the mission and the size and geographical span of the network, the issue of transporting the observations and measurements to the right aggregation points in time, the overhead cost of transmission and processing, potential attacks on measurements and on the tradeoffs—all can pose significant challenges.

Another direction that is worthwhile to explore stems from the fact that this framework is self-referential and reflective in that it measures itself (i.e., measurements and observations about the system) against the yardsticks that was set for the specific system (i.e., the requirements laid out by the stakeholders for the mission). While this provides a good way to assess IA within a system as long as the metrics and levels are used consistently for all stakeholders, extending this to multiple systems can be problematic because the systems may not provide the exact set of metrics. To support comparative assessment or cross comparison of multiple systems further investigation of metric classes exploring equivalence and other calibration relationships is needed.

6.0 REFERENCES

[1] Avizienis et al. "Fundamental Concepts of Dependability," ISW-2000, Cambridge, MA, 2000.

[2] Chong et al. “Survivability Architecture of a Mission Critical System: The DPASA Example,” ACSAC-21, 2005.

[3] Benjamin et al. “Using A Cognitive Architecture to Automate Cyber Defense Reasoning,” ECIS BLISS Symposium, Edinburgh, 2008.

[4] Pal et al. "The DPASA Survivable JBI- A High-Water Mark in Intrusion-Tolerant Systems", WRAITS workshop, Lisbon, 2007.

[5] Loyall et al. “Specifying and Measuring QoS in Distributed Object Systems,” Proc of ISORC, Kyoto, 1998.

[6] Rohloff et al. "Scalable, Distributed, Dynamic Resource Management for the ARMS Distributed Real-Time Embedded System." WPDARTS 2007.

[7] Atighetchi et al. “Adaptive Cyberdefense for Survival and Intrusion Tolerance.” IEEE Internet Computing, Vol. 8, No. 6, November/December 2004, pp. 25-33.

[8] Nagios Development Team and Community Contributors. “Nagios Core Version 3.x Documentation.” *Nagios*®. http://nagios.sourceforge.net/docs/3_0/ NAGIOS, 2009

[9] Hay et al. OSSEC HIDS Guide. Massachusetts: Syngress Publishing, Inc., 2008.

[10] CVE: http://cve.mitre.org/.

[11] CVSS: http://www.first.org/cvss/.

[12] SCAP: http://scap.nist.gov/.

[13] http://csrc.nist.gov/publications/nistpubs/800-55-Rev1/SP800-55-rev1.pdf

[14] http://csrc.nist.gov/publications/fips/fips1403/fips1403-Draft.pdf

[15] CC: http://www.commoncriteriaportal.org/thecc.html

[16] CWE: http://cwe.mitre.org/.

[17] Red team: http://idart.sandia.gov/index.html

[18] Foley et al. “Multilevel Security and Quality of Protection,” ACM QoP workshop, 2007.

[19] Aime et.al. “AMBRA: Automated Model-Based Risk Analysis,” ACM QoP workshop, 2007.

[20] D. Maughan, “The need for a national cybersecurity research and development agenda,” CACM, vol. 55, no. 2, Feb 2010, pp.29-31.