Metrics of Security

Yi Cheng,

Julia Deng,

Jason Li,

Scott DeLoach,

Anoop Singhal,

Xinming Ou

1 Introduction

Discussion of challenges and ways of improving Cyber Situational Awareness dominated previous chapters in this book. However, we have not yet touched on how to quantify any improvement we might achieve. Indeed, to get an accurate assessment of network security and provide sufficient Cyber Situational Awareness (CSA), simple but meaningful metrics – the focus of the Metrics of Security chapter – are necessary. The adage, "what can't be measured can't be effectively managed," applies here. Without good metrics and the corresponding evaluation methods, security analysts and network operators cannot accurately evaluate and measure the security status of their networks and the success of their operations. In particular, this chapter explores two distinct issues: (i) how to define and use metrics as quantitative characteristics to represent the security state of a network, and (ii) how to define and use metrics to measure CSA from a defender's point of view.

To provide sufficient CSA and ensure mission success in enterprise network environments, security analysts need to continuously monitor network operations and user activities, quickly identify suspicious behaviors and recognize malicious activities, and mitigate potential cyber impacts in a timely manner. However, most existing security analysis tools focus on detecting attacks. The massive amounts of security-related data make these approaches not only labor intensive, but also too prone to error while providing users a "big picture" of their overall cyber situation. Security analysts need more sophisticated and systematic methods to quantitatively evaluate network vulnerabilities, predict attack risk and potential impacts, assess proper actions to minimize business damage, and ensure mission success in a hostile environment. As a natural requirement, security metrics are very important for CSA, coordinated network defense, and mission assurance analysis. They can provide a better understanding of the adequacy of security controls, and help security analysts effectively identify which critical assets to focus their limited resources on in order to ensure mission success.

For CSA and mission assurance analysis, security metrics need to be aligned not only with the industry standards for computer and network security management, but also with the overall organizational and business goals in enterprise environments. This chapter discusses the methodology to effectively identify, define, and apply simple but meaningful metrics for comprehensive network security and mission assurance analysis. Focusing on enterprise networks, we will explore security tools and metrics that have been developed, or need to be developed, to provide security and mission analysts the capabilities required to better understand the cyber situation and security status of their network. For instance, is there any vulnerability on the system? Is there any (ongoing) attack in the network? What (system/application/service) has been compromised? How can the (potential) risk be measured? What is the most likely consequence of the attack? Can we prevent it? How much (storage/communication/operational) capacity will be lost due to the attack? Is the overall (or a major portion of) mission/task/operation still accomplished? Good defined metrics can help users answer these questions quickly and quantitatively. Users can then focus on the higher-level view of cyber situations, make informed decisions to select the best course of action, effectively mitigate the potential threats, and ensure mission success even in a hostile environment.

Researchers have made many attempts during the last few years to measure security in cyberspace. NIST provided an overview of existing metrics for network security measurement in (Jansen, 2009). Hecker (2008) distinguished the lower level metrics (based on well-ordered low-level quantitative system parameters) from the higher level metrics (e.g., conformity distance, attack graph or attack surface based estimations). Meland and Jensen (2008) presented a Security-Oriented Software Development Framework (SODA) to adapt security techniques and filter information. Heyman et al. (2008) also presented their work on using security patterns to combine security metrics.

In this chapter we present a new model for security risk analysis. Section 2 present a model for Cyber Situational Awareness. Section 3 discusses Network Vulnerability and section 4 discusses Mission Impact Analysis. Section 5 discusses Asset Criticality Analysis. Section 6 presents some ideas for future work and finally section 7 gives a summary for this chapter.

2 Security Metrics for Cyber Situational Awareness

2.1 Security Metrics: the What, Why, and How

2.1.1 What is a Security Metric?

As defined by the National Institute of Standards and Technology (NIST), metrics are tools that are designed to facilitate decision-making and improve performance and accountability through collection, analysis, and reporting of relevant performance-related data. Security metrics can be naturally interpreted as a standard (or system) used for quantitatively measuring an organization's security posture. Security metrics are essential to comprehensive network security and CSA management. Without good metrics, analysts cannot answer many security related questions. Some examples of such questions include "Is our network more secure today than it was before?" or "Have the changes of network configurations improved our security posture?"

The ultimate aim of security metrics is to ensure business continuity (or mission success) and minimize business damage by preventing or minimizing the potential impact of cyber incidents. To achieve this goal, organizations need to take into consideration all information security dimensions, and provide stakeholders detailed information about their network security management and risk treatment processes.

2.1.2 Why Security Metrics for CSA?

We cannot effectively manage or improve CSA if we cannot accurately measure it. Traditional network security management practices mainly focus on the information level and treat all network components equally. Although valuable, these approaches lack meaningful metrics and risk assessment capabilities when applied to comprehensive CSA and mission assurance analysis. Specifically, they cannot quantitatively evaluate or determine the exact impacts of security incidents on the attainment of critical mission objectives. When an attack happens, it is really difficult for current solutions to answer mission assurance related security questions like: "Is there any impact on mission X if host A was compromised?", "Can some portion of mission X still be accomplished?", "What is the average completion rate for mission X currently?", or "What can we do to ensure mission X's success?"

To answer these questions, security metrics and advanced mission-to-asset mapping, modeling and evaluation technologies are required. The literature contains several recently proposed metrics for information and network security measurement, such as the number of vulnerabilities or detected cyber incidents in a network, the average response time to a security event, etc. Although these metrics can evaluate network security from certain aspects, they cannot provide sufficient network vulnerability assessment, attack risk analysis and prediction, mission impact mitigation, and quantitative situational awareness, in terms of mission assurance. We argue that to ensure mission survival in a hostile environment, ideally security metrics should be adjusted and tuned to fit a specific organization or situation. In other words, good metrics must be meaningful to specific organizational goals and key performance indicators. Security analysts not only review metrics currently in place, but also need to ensure they are aligned with the specific organizational and business goals.

2.1.3 How to Measure and Model Network Security?

To determine the general security level of an analyzed network, a common process needs to be realized: First, security experts identify what should be measured. Then they organize the involved variables in a manageable and meaningful way. After that, repeatable formulas should be built to illustrate the snapshot status of security and how it changes over time. For network and/or system security measurement, most existing approaches are based on *risk analysis*, in which security risk is expressed as a function of threats, vulnerabilities, and potential impacts (or expected loss).

$$Risk = Threat \times Vulnerability \times Impact \qquad (1)$$

Eq. 1 is an informal way of stating that security risk is a *function* of threats, vulnerabilities, and potential impact. It is often used in the literature for expressing the necessity and purpose of network security evaluation. When applied to solving a real problem, it is still hard to quantify each variable in Eq. 1 with meaningful values. For example, how should one numerically express a threat? What is the cost of a vulnerability? How should one calculate the impact or expected loss? In addition, when these three variables are multiplied, how should one denote the risk in a meaningful way that can be translated into an action item?

In order to quantify different portions of Eq. 1, Lindstrom (2005) further introduced a number of underlying elements required for general security (risk) analysis. Although they may not completely solve all the problems, these underlying elements still provide security analysts a better understanding and insight to develop meaningful metrics and practical solutions for general network security measurements. Some of the useful elements introduced by (Lindstrom, 2005) are listed below:

- **Calculation of Asset Value:** Based on the values of different assets (e.g., hardware, software and data), enterprises can focus on their real security needs and allocate adequate resources. As enterprises routinely place values on their information assets, the value of an asset could be defined as the amount of IT spending over a time period (e.g., operations and maintenance) plus the depreciation or amortization value of the assets (hardware and software). For asset value calculation, quantifiable values need to be assigned to each asset for objective evaluation and comparison.
- **Calculation of Potential Loss:** Asset value is linked, but not tied directly to the loss. We need to consider the type of compromise when evaluating the potential losses. Generally there are five distinct types of compromise: *confidentiality breaches*, *integrity breaches*, *availability breaches*, *productivity breaches*, and *liability breaches* (Lindstrom, 2005). Note that asset value may not be the only thing that can be lost. Other potential losses, such as the incident costs should also be carefully considered.
- **Measurement of Security Spending:** Although measuring enterprise-wide security spending is difficult, it is important for security management. Security spending is often divided among various business units and departments, as well as being lumped in with network and infrastructure spending. Assessing security spending and separating it from other budget items is a daunting task.
- **Attack Risk Analysis:** Defining and modeling risk for an enterprise is another difficult but important task. Lindstrom (2005) lists three common forms of risks: *manifest risk* (the ratio of malicious events to total events), *inherent risk* (the likelihood that system configurations will contribute to a compromise), and *contributory risk* (a measure of process errors or mistakes made during the operations).

None of the above elements is designed to completely answer questions related to security metrics and measurements, but the methodologies outlined here give us a foundation for gathering useful data and applying it to our specific goals and expectations. Based on this basic knowledge, researchers can further define more accurate and complete security metrics, assign proper values to their security formulas, and develop practical evaluation models to quantitatively analyze and measure the security status of their computer network and systems.

2.2 Security Measurement for Situational Awareness in Cyberspace

Generally speaking, security measurement for CSA needs to carefully consider two distinct possible issues: (i) How to define and use metrics as quantitative characteristics to represent the security state of a computer system or network, and (ii) How to define and use metrics to measure CSA from a defender’s point of view. This section will briefly review state-of-the-art security metrics and discuss the challenges to define and apply good metrics for comprehensive CSA and mission assurance analysis.

2.2.1 Quantification and Measurement of Traditional Situational Awareness

A general definition of Situational Awareness (SA) is given by (Endsley, 1988): “SA is the perception of the elements of the environment within a volume of time and space, the comprehension of their meaning, and the projection of their status in the near future.” Due to its multivariate nature, a considerable challenge is posed for SA quantification and measurement. Traditional SA measurement techniques can be generally considered either based on “product-oriented” direct measurement (e.g., objective real-time probes or subjective questionnaires assessing perceived SA), or the “process-oriented” inference of operator behavior or performance (Fracker, 1991a) (Fracker, 1991b).

According to (Bolstad C. and Cuevas H., 2010), existing SA measurement approaches can be further classified into the following categories:

- **Objective Measures**: Comparing an individual’s perceptions of the situation or environment to some “ground truth” reality (Jones D. and Endsley M. R., 2000). This type of assessment provides a direct measure of SA and does not require operators or observers to make judgments about situational knowledge on the basis of incomplete information. Generally, objective measures can be gathered in three ways: (i) in real-time as the task is completed, (ii) during an interruption in task performance, or (iii) post-test following completion of the task (Endsley, 1995).
- **Subjective Measures:** Asking individuals to rate their own or the observed SA of individuals on an anchored scale (Strater L., et al., 2001). Subjective measures of SA are relatively straightforward and easy to administer, but they also suffer from several limitations. For example, individuals are often unaware of information they do not know, and they cannot fully exploit the multivariate nature of SA to provide detailed diagnostics (Taylor, 1989).

- **Performance Measures:** Assuming that better performance usually indicates better SA, performance measures infer SA from performance outcomes. Bolstad and Cuevas (2010) list a set of commonly used performance metrics, including the quantity of output or productivity level, time to perform the task or respond to an event, the accuracy of the response, and the number of errors committed. In addition, good SA does not always lead to good performance, and poor SA does not always lead to poor performance (Endsley, 1990). Performance measures should be used in conjunction with others measures for more accurate assessment.
- **Behavioral Measures:** Based on the assumption that good actions usually follow from good SA and vice-versa, behavioral measures infer SA from individuals' actions. Behavioral measures are subjective in nature, as they primarily rely on observer ratings. To reduce this limitation, observers need to make judgments based on good SA indicators that are more readily observable (Strater L., et al., 2001), (Matthews M., et al., 2000).

Note that the multivariate nature of SA significantly complicates its quantification and measurement. A particular metric may only tap into one aspect of the operator's SA. Durso et al. (1995), Endsley et al. (1998), and Vidulich (2000) also found that different types of SA measures do not always correlate strongly with each other. In this case, multi-faceted approaches that combine distinct but highly related measures should be used for comprehensive SA measurement, as they can take advantage of the strengths of each measure while minimizing their inherent limitations (Harwood K., et al., 1988).

2.2.2 State-of-the-Art Security Measurement Techniques

Researchers have made many attempts during the last few years to measure SA in cyberspace. NIST provided an overview of existing metrics for network security and SA measurement in (Jansen, 2009). Hecker (2008) distinguished the lower level metrics (based on well-ordered low-level quantitative system parameters) from the higher level metrics (e.g., conformity distance, attack graph or attack surface based estimations). Meland and Jensen (2008) presented a Security-Oriented Software Development Framework (SODA) to adapt security techniques and filter information. Heyman et al. (2008) also presented their work on using security patterns to combine security metrics.

To define software security metrics, Wang et al. (2009) proposed a new approach based on vulnerabilities in the software systems and their impacts on software quality. They used Common Vulnerabilities and Exposures (CVE) (http://cve.mitre.org/cve/) and Common Vulnerability Scoring System (CVSS) (http://www.first.org/cvss/) in their metric definition and calculation. An attack surface based metric was further proposed by Manadhata and Wing (2011) to measure software security. They formalized the notion of a system's attack surface, and used it as an indicator of the system's security. By measuring and reducing attack surfaces, software developers can effectively mitigate their software's security risks.

Petri nets (PN) have also been discussed as a useful formalism for network security evaluation in literature. The idea of using PN for attack analysis was first introduced by McDermott (2000).

Several papers consider the use of Colored PN (CPN) for attack modeling. Zhou et al. (2003) discussed the advantages of CPNs and described a process for mapping an attack tree to a CPN. Dahl (2005) provided a more detailed discussion of the advantages of CPN when it was applied to model concurrency and attack progress.

For CSA and risk assessment in enterprise networks, an ontology-based Cyber Assets to Missions and Users (CAMUS) mechanism was proposed by Goodall (2009). It can automatically discover the relationship between cyber assets, missions and users to facilitate cyber incident mission impact assessment. The basic idea of CAMUS came from Air Force Situation Awareness Model (AFSAM) (Salerno, 2008) (Salerno J., et al., 2005), which described how data is taken to become information and consumed by analysts to further improve the situation management. Tadda et al. (2006) refined the general AFSAM and applied it directly to the cyber domain, resulting in the CSA model. Within the CSA model, the knowledge required for situation management is an accurate understanding of how operations are impacted when there are degradations and compromises in the cyber infrastructure. Grounded in the CSA model, Holsopple et al. (2008) developed a Virtual Terrain that models the network by manually taking mission context into account.

Grimaila et al. (2008) shifted their focus to information asset situation management. They proposed a Cyber Damage Assessment Framework that requires the manual definition and prioritization of both operational processes and information assets. Gomez et al. (2008) proposed an approach for automated assignment of intelligence, surveillance and reconnaissance (ISR) assets to specific military missions. Their Missions and Means Framework (MMF) ontology includes similar concepts in CAMUS, such as missions, operations, tasks, capabilities and systems. Lewis et al. (2008) also proposed a mission reference model to tackle the mapping of cyber assets to missions, based on a mathematical constraint satisfaction approach.

To support enterprise level security risk analysis, Singhal et al. (2010) provided a security ontology framework as a portable and easy-to-share knowledge base. Based on this framework, analysts will know which threats endanger which assets and what countermeasures can lower the probability of the occurrence of an attack. Alberts et al. (2005) proposed a risk-based assessment protocol, called Mission Assurance Analysis Protocol (MAAP), to qualitatively evaluate current conditions and determine whether a project or process is on track for success. MAAP can produce a rich, in-depth view of current conditions and circumstances affecting a project's potential success, but its risk assessment is a complex and time-consuming process. Watters et al. (2009) proposed a Risk-to-Mission Assessment Process (RiskMAP) to connect business objectives to network nodes. RiskMAP first models key features of a corporation (from business objectives, operational tasks, information assets, to network nodes that store, send and make the information available), and then uses the same model to map network level risks to the upper level business objectives for risk analysis and impact mitigation.

Musman et al. gave an outline of the technical roadmap for mission impact assessment in a MITRE report (2010). They focused on cyber mission impact assessment (CMIA) and tried to link network and information technology (IT) capabilities to an organization's business processes (missions). Grimaila et al. (2009) discussed general design concepts of a system that provides the decision makers with notifications on cyber incidents and their potential impacts on missions. Several approaches based on attack graphs were also investigated for automated attack detection and risk analysis (Noel S., et al., 2004) (Qin X. and Lee W., 2004) (Cheung S., et al., 2003).

Jakobson (2011) further proposed a logical and computational attack model for cyber impact assessment. In his framework, a multi-level information structure, called "cyber-terrain," was introduced to represent cyber assets, services, and their inter-dependencies. The dependencies between the cyber terrain and missions are represented by an impact dependency graph. Using these graphical models, both direct impacts and the prorogation of cyber impacts on missions through the inter-connected assets and services can be calculated. In (Kotenko I., et al., 2006), the authors proposed a new approach for network security evaluation, based on comprehensive simulation of malefactors' actions, construction of attack graphs, and computation of different security metrics. A software tool was offered for vulnerability analysis and security assessment at various stages of a life cycle of computer networks.

2.2.3 Security Measurement for Enterprise CSA: Challenges & Potential Solutions

State-of-the-art technologies provide useful descriptive information on security analysis, mission modeling, and situation management. While they are quite valuable for security measurement in various situations, existing approaches still face several challenges when applied to CSA and mission assurance assessment in enterprise network environments, due to the lack of meaningful security metrics and efficient evaluation methods.

Briefly speaking, existing methods have suffered from the following limitations that reduce their usefulness and effectiveness for CSA and mission assurance analysis:

- Lack of real-time CSA
- Lack of understanding of impacts of cyber events on high level mission operations
- Lack of quantitative metrics and measures for comprehensive security assessment
- Lack of incorporating human (analyst) cognition into cyber-physical situational awareness
- Lack of mission assurance policy

Table 1 compares current technologies and systems developed for mission asset mapping and modeling, cyber-attack and intrusion detection, risk analysis and prediction, as well as for damage assessment and mission impact mitigation. Each method has its own strength and limitations. When applied for enterprise network CSA, mission assurance assessment and coordinated network defense, advance technologies, mathematic models and evaluation algorithms are still required to answer the following questions:

- How to identify and represent mission composition and dependency relationships?
- How to derive the dependency relationships between mission elements and cyber assets?

- As a single vulnerability may enable widespread compromises in an enterprise, how to quickly identify the start point of an attack and predict its potential attack path?
- How to assess the direct impact and propagation of cyber incidents on high level mission elements and operations?
- How to systematically represent and model the identified inter- and intra- dependency relationships between major elements or components involved in cyber mission assurance?
- How to define and develop quantitative metrics and measures for meaningful cyber situational awareness, enterprise security management and mission assurance analysis?

Table 1: Limitations of Existing Approaches for CSA

Approach	Technology Strength	Developer	Limitations
CAMUS	Ontology fusion based cyber assets to missions and users mapping	Applied Visions, Inc.	Centralized approach Lack of cyber impact assessment Lack of mission asset prioritization
MAAP	Mission assurance and operational risk analysis in complex work processes	Carnegie Mellon University	Centralized approach Focus on operational risk analysis Lack of mission asset dependencies
RiskMAP	Risk-to-mission assessment at network and business objectives levels	MITRE	Centralized approach Lack of mission asset dependencies
Ranked Attack Graph	Identifying critical assets based on page rank and reachability analysis on attack graphs	Carnegie Mellon University	Lack of mission models Cannot analyze cyber impacts on high level missions
CMIA	Cyber mission impact assessment based on military mission models	MITRE	Centralized approach Lack of cyber impact analysis; Lack of mission asset prioritization

To address these challenges, key technologies such as quantitative and meaningful security metrics need to be further investigated and developed. In this chapter, we will introduce some potential solutions and results of our initial study that leverages and extends recent advances in CSA, mission assurance, common vulnerability assessment, and enterprise security management. As a starting point, our study focuses on developing an integrated framework for real-time CSA and mission assurance analysis in enterprise environments. To achieve this objective, a group of simple but meaningful metrics and corresponding evaluation methods were investigated for three specific use cases: (i) network vulnerability and attack risk assessment, (ii) cyber impact and mission relevance analysis, and (iii) asset criticality analysis and prioritization.

Table 2 lists a set of security and performance metrics, mainly focusing on network vulnerability assessment, attack risk evaluation, and mission impact analysis. Each metric defined in Table 2 attempts to answer a specific question related to computer/network security, system performance, or mission assurance. For instance, the *Vulnerable Host Percentage* (*VHP*) metric tries to answer how many hosts could be compromised in the worst case. The *Average Length of Attack Paths* (*ALAP*) metric attempts to answer the typical effort required for an attacker to violate a security policy. Obviously, each metric has shortcomings if only used by itself for network security analysis. For example, the *Shortest Attack Path* (*SAP*) metric ignores the number of ways an attacker may violate a security policy; the *ALAP* metric fails to adequately account for the number of ways an attacker may violate a security policy; while the *Number of Attack Paths* (*NAP*) metric ignores the effort associated with violating a security policy. Therefore, multiple security metrics must be used together to provide users with a comprehensive view and understanding of cyber situational awareness and mission assurance.

Table 2: Common Security and Performance Metrics for CSA

Metric	Acronym	Description	Score/Value
Asset Capacity	AC	The (remained) capacity of a cyber asset (after being attacked or compromised)	[0, 1]: 0 means not operational; 1 means fully operational
Average Length of Attack Paths	ALAP	The average effort to penetrate a network, or compromise a system/service; evaluated by attack graphs	n: the average length of potential attack paths
Compromised Host Percentage	CHP	The percentage of compromised hosts in a network at time *t*	[0, 1]: 0 means no compromise; 1 means all compromised
Exploit Probability	EP	How easy (or hard) to exploit a vulnerability? Could be measured by CVSS exploitability sub-score	[0, 1]: 0 means hard to exploit; 1 means easy to be exploited
Impact Factor	IF	The impact level of a vulnerability after being exploited, could be measured by CVSS impact sub-score	[0, 1]: 0 means no impact; 1 means totally destroyed
Number of Attack Paths	NAP	The number of potential attack paths in a network, could be evaluated based on attack graphs	n: the number of potential attack paths
Network Preparedness	NP	Is a network ready to carry out a mission? E.g., all required services are supported by available cyber assets	[0, 1]: 0 means not ready; 1 means fully ready
Network Resilience	NR	The percentage of compromised systems/services that can be replaced/recovered by backup/alternative systems/services	[0, 1]: 0 means cannot recover; 1 means can be fully recovered
Operational Capacity	OC	The (remained) operational capacity of a system/service (after being affected by a direct attack or indirect impact)	[0, 1]: 0 means not operational; 1 means fully operational
Resource Redundancy	RR	Is there any redundant (backup) resources assigned or allocated for a critical task/operation?	0 or 1: 0 means no backup system; 1 means at least 1 backup system
Service Availability	SA	The availability of a required service to support a particular mission, task, or operation	0 or 1: 0 means not available; 1 means service is available
Shortest Attack Path	SAP	The minimal effort to penetrate a network, or compromise a system or service, evaluated by attack graphs	n: the shortest length of potential attack paths
Severity Score	SS	The severity/risk of a vulnerability if it was successfully exploited, could be measured based on CVSS score	[0, 1]: 0 means no risk; 1 means extremely high risk
Vulnerable Host Percentage	VHP	The percentage of vulnerable hosts in a network	[0, 1]: 0 means no vulnerable host; 1 means all hosts are vulnerable

Note that the security and performance metrics, as well as the corresponding evaluation mechanisms introduced in this chapter are not trying to completely solve enterprise CSA quantification and measurement problems. The objective here is to help security analysts to have a better understanding and insight to further develop their own good and meaningful metrics, as well as practical solutions, for their specific questions related to CSA, mission assurance, or enterprise network security defense.

3 Network Vulnerability and Attack Risk Assessment

Although the ultimate goal for enterprise network security goal is to identify and remove all network and host vulnerabilities, it is infeasible to achieve this goal in practice. For instance, if an organization leverages Commercial-Off-the-Shelf (COTS) software to operate its network, it will expose itself to the vulnerabilities that the software possesses. Issues such as slow and unstable released patches may cause the organization to operate its network with known vulnerabilities. Through these vulnerabilities, attackers may successfully compromise a particular system via a single attack action, or penetrate a network via a series of attack actions. Therefore, network vulnerability and attack risk assessment is the first step for enterprise security management and cyber situational awareness.

3.1 Security Metrics for Vulnerability Assessment

3.1.1 Common Vulnerability Assessment on Computer System

In literature, the Common Vulnerability Scoring System (CVSS) (http://www.first.org/cvss/) has been widely adopted as the primary method for assessing the severity of computer system security vulnerabilities. As an industry standard, CVSS ensures repeatable accurate measurement. It also enables users to see the underlying vulnerability characteristics that were used in its quantitative models to generate the scores. CVSS attempts to establish a measure of how much concern a vulnerability warrants compared to other vulnerabilities. It is composed of three metric groups: *Base*, *Temporal*, and *Environmental*. Each group consists of a set of metrics, as shown in Fig. 1.

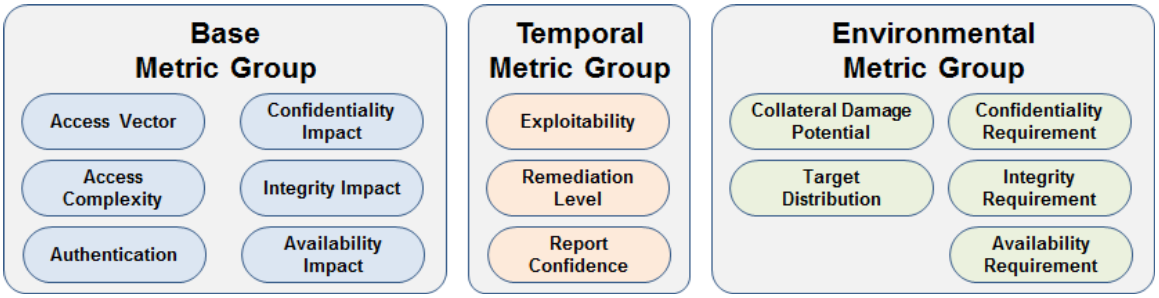


Fig. 1: CVSS Metric Groups (http://www.first.org/cvss/cvss-guide)

In particular, base metrics define criticality of the vulnerability, temporal metrics represent urgency of the vulnerability that changes over time, and environmental metrics represent the characteristics of a vulnerability that are relevant and unique to a particular user's environment. Each group produces a numeric score (ranging from 0 to 10) and a compressed textual representation that reflects the values used to derive the score. The CVSS complete guide (http://www.first.org/cvss/cvss-guide) gives the detailed descriptions of these metric groups:

- **Base:** representing “intrinsic and fundamental characteristics of a vulnerability that are constant over time and user environments,”
- **Temporal:** representing “characteristics of a vulnerability that change over time but not among user environments,” and
- **Environmental:** representing “characteristics of a vulnerability that are relevant and unique to a particular user's environment.”

Basically, for each metric group, a particular equation is used to weigh the corresponding metrics and produce a score (ranged from 0 to 10) based on a series of measurements and assessments of security experts, with the score 10 representing the most severe vulnerability. Specifically, when the base metrics are assigned values, the base equation calculates a score ranging from 0 to 10, and creates a vector. This vector is a text string that contains the values assigned to each metric, and facilitates the “open” nature of the framework. Users can understand how the score was derived and, if desired, confirm the validity of each metric. More details on base, temporal and environmental equations, as well as the calculation methods can be found in the CVSS complete guide (http://www.first.org/cvss/cvss-guide).

3.1.2 General Metrics for Network Vulnerability Assessment

The National Vulnerability Database (NVD) (http://nvd.nist.gov/) provides CVSS scores for almost all known vulnerabilities. Various open source or commercial vulnerability scanners, such as Nessus Security Scanner (http://www.tenable.com/products/nessus), Open Vulnerability Assessment System (OpenVAS) (http://www.openvas.org/), and Microsoft Baseline Security Analyzer (MBSA) (http://www.microsoft.com/en-us/download/details.aspx?id=7558), can be used to feasibly identify vulnerabilities in a network. Regularly and periodically performing vulnerability scan and assessment is critical for enterprise security management, as it can easily locate which systems are vulnerable, identify what services/components are vulnerable, and suggest the best method for repairing the vulnerabilities before attackers find and exploit them. To evaluate the general security of an enterprise network based on vulnerability assessment, we use three security metrics: the vulnerable host percentage (VHP), CVSS severity score, and compromised host percentage (CHP).

1) The Vulnerable Host Percentage (VHP)

This metric represents the overall security level of a network. The number of vulnerable hosts can be obtained by periodically scanning a network via vulnerability scanning tools such as Nessus. The equation for this metric is given below, where *G* represents an intended network, *V* is the set of vulnerable hosts, and *H* is the set of all hosts in the network.

$$VHP(G) = 100 \times \frac{|\{v{:}v \in V \subseteq H\}|}{|\{h{:}h \in H\}|} \qquad (2)$$

2) Severity Score of a single vulnerability i (SS_i)

After identifying vulnerabilities that exist in a network, we need to know the severity score of each identified vulnerability based on CVSS. As shown in Table 3, this metric indicates the severity of certain vulnerability, and how to handle it accordingly.

Table 3: Severity Levels of Vulnerabilities

CVSS Score	Severity Level	Guidance
7.0 through 10.0	High Severity	Must be corrected with the highest priority
4.0 through 6.9	Medium Severity	Must be corrected with high priority
0.0 through 3.9	Low Severity	Encouraged, but not required, to correct these vulnerabilities

3) Compromised Host Percentage (CHP)

This metric indicates the percentage of hosts that have been compromised in a network. Here, a host compromise is defined as the attacker having obtained the user- or administrator- level privilege on the intended host. Higher CHP value means more hosts are compromised. Our general goal is to minimize the CHP metric. For instance, an organization should have stricter firewall rules and user access policies so that it is hard to exploit the vulnerabilities (from both outside and inside). The equation for this metric is given below, where *G* represents an intended network, *C* is the set of compromised hosts, and *H* is the set of all hosts in the network.

$$CHP(G) = 100 \times \frac{|\{c : c \in C \subseteq H\}|}{|\{h : h \in H\}|} \qquad (3)$$

3.1.3 Attack Graph based Network Vulnerability Assessment

In cyberspace, attackers may successfully compromise a particular system via a single attack action, or penetrate a network via a series of attack actions. A series of attack actions is usually referred to as a multi-step attack or chained exploit. It leverages the interdependencies among multiple vulnerabilities to violate a network's security policy. In the literature, the multi-step attack can be feasibly represented and modeled via various attack graph models (Ou X., et al., 2006) (Sheyner O., et al., 2002) (Ammann P., et al., 2002). Attack graphs is a widely adopted technology in analyzing the casual relationships between cyber-attack events, in which each node represents a particular state of a cyber asset in a network, and each edge represents a possible state transition. In our framework, attack graph based metrics are also defined for network-level vulnerability assessment.

1) The Number of Attack Paths (NAP)

This metric indicates how many ways an attacker can penetrate the network or compromise a critical system. The equation for this metric is given below, where *G* represents an intended network, *AG* represents the derived attack graphs, and *P* is the set of potential attack paths in the derived attack graphs.

$$NAP(G) = |\{p: p \in P \subseteq AG\}| \qquad (4)$$

2) The Average Length of Attack Paths (ALAP)

This metric represents the average amount of effort that an attacker needs to take in order to penetrate the network or compromise a critical system. The equation for this metric is given as below, where *L*(*p*) represents the length of attack path *p*.

$$ALAP(G) = \frac{|\{L(p):p\in P\subseteq AG\}|}{|\{p:p\in P\subseteq AG\}|} \qquad (5)$$

3) The Shortest Attack Path (SAP)

This metric indicates the least amount of effort that an attacker can take to penetrate the network or compromise a critical system. The metric is given as below.

$$SAP(G) = \min\{L(p)\ |p \in P \subseteq AG\} \qquad (6)$$

3.2 Modeling and Measurement of Attack Risk

3.2.1 Attack Risk Prediction

To quantitatively evaluate cyber impacts on high level missions, mission related elements such as cyber assets, hardware devices, and mission tasks should be added to the risk analysis model. Leveraging the basic analysis method and evaluation process proposed by Jakobson (2011), we extend our attack risk prediction model with cyber assets, hardware device, and mission elements in our initial study. We believe this model can be used to quantitatively evaluate the severity of an identified vulnerability, and analyze the consequence if a mission critical asset was attacked or compromised. Using our initial study as the starting point, more complete and concrete analysis models can be further developed.

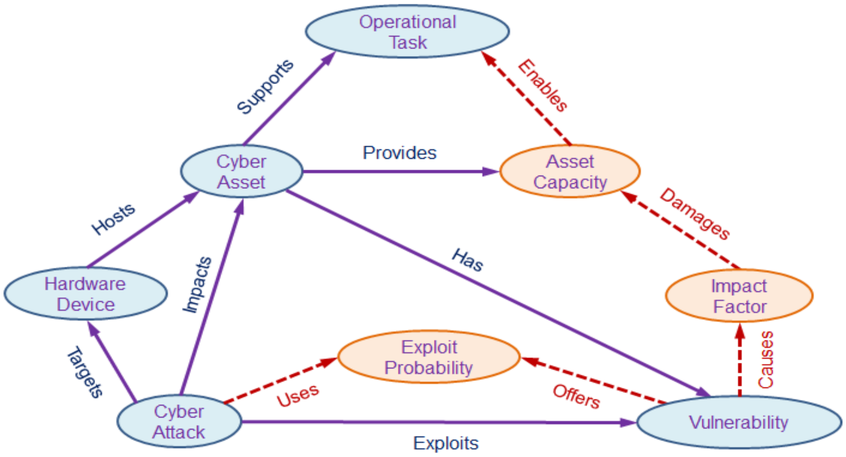


Fig. 2: Attack Risk Prediction Model for Mission Impact Analysis

Our initial study focused on modeling (i) the logical relations that allow us to model the propagation of the impacts through the network, and (ii) the computational relations that allow us to calculate the level of those impacts. The conceptual structure of the extended attack model is illustrated by Fig. 2. It contains eight conceptual nodes: *Cyber Attack*, *Hardware Device*, *Cyber Asset*, (Asset) *Vulnerability*, *Operational Task*, *Asset Capacity*, *Exploit Probability* and *Impact Factor* (of Vulnerability), as well as the corresponding relations among them.

As pointed by Jakobson (2011), the *Exploit Probability* (*EP*) and *Impact Factor* (*IF*) of the vulnerability, as well as the *Asset Capacity* (*AC*) of the asset, are important parameters in attack risk analysis. Specifically, *EP* is a measure defined in an interval [0, 1], which indicates to what degree the vulnerability can be exploited to compromise the attacked asset. For instance, *EP = 0* means that this vulnerability is effectively impossible to exploit (hence the attack has no impact on the target asset). Conversely, *EP = 1* means that the vulnerability is easy to exploit to compromise the intended asset. The *Impact Factor* (*IF*), on the other hand, indicates how much damage can be caused by an attack. It is also a measure defined in an interval [0, 1]. *IF = 0* means that the attack has no impact on an asset, while *IF = 1* means that an asset can be totally destroyed (i.e., lose all of its capacity).

3.2.2 Damage Assessment

The *Asset Capacity* (*AC*) is another important measure to characterize the operational capacity of a cyber asset. It indicates how much capacity an asset can still provide to fulfill its function after being attacked. In our model, *AC* can be measured in an interval [0, 1]. Value 0 means the asset is not operational at all; while value 1 means that the asset is fully operational. Note that the computational relation between *EP*, *IF* and *AC* allows us to calculate and measure how the capacity of an asset could be affected by an attack, which further enables the quantitative analysis of the mission impacts caused by the attack.

According to (Jakobson G., 2011), the general calculation of mission impacts should contain the following steps:

1) **Attack Start Point Detection:** The first step is to identify the start point of an attack. Currently, we use leaf nodes in our attack graphs as the start points.
2) **Direct Impact Assessment:** The next step is to determine the direct impact of an attack on the targeted asset. We follow the extended attack model in Fig. 2 and calculate the direct impact based on CVSS.
3) **Propagation of Cyber Impacts through the Network:** In this step, we calculate the potential impacts on cyber capacities of all mission-related assets along the attack paths derived from our attack graphs.

4) **Mission Impact Assessment:** After knowing the current capacities of all assets involved in a mission, we can further calculate the potential impacts on the high level missions based on mission asset dependency relationships derived by our logical mission models.

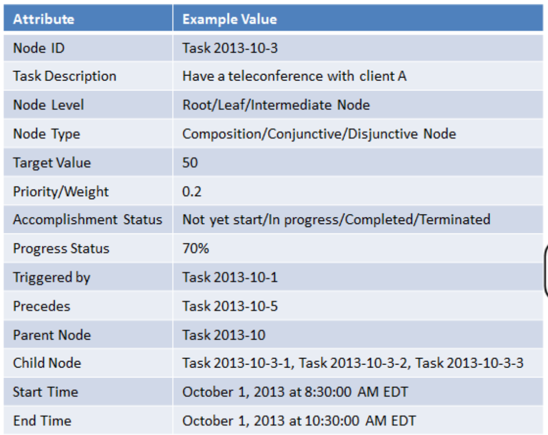
It should be noted that figuring out how to assign the proper value to *EP* and *IF* could be a critical task that requires analysis of historical attack data as well as consultation with cyber security experts. In our initial study, the *Exploitability Score* (*ES*) and *Impact Score* (*IS*) in CVSS have been used as our starting point to calculate *EP* and *IF*. As both *ES* and *IS* range from 0 to 10 in CVSS, we calculate these two parameters by: $EP = ES/10$, and $IF = IS/10$.

4 Cyber Impact and Mission Relevance Analysis

Impact assessment is important for mission assurance analysis in cyberspace, where critical mission elements must rely on the support of the underlying cyber network and compromised assets may have significant impacts on a mission's accomplishment. As described in previous sections, for cyber mission assurance assessment, we need practical analysis models to effectively represent a complex mission, and the dependency relationship between high level mission elements and the underlying cyber assets. We also need to build a mission impact propagation model to investigate the direct and indirect consequences caused by malicious cyber incidents on high level mission elements and tasks. In addition, quantitative metrics and measures are required for meaningful mission assurance and cyber situational awareness analysis.

4.1 Mission to Asset Mapping and Modeling

To efficiently represent and model the dependency relationships between high level mission elements and the underlying computer network and cyber assets, a Logical Mission Model (LMM) is developed in our framework. Essentially, the LMM is a hierarchical graphical model for mission planning, decomposition, modeling, and asset mapping, which is further composed of a Value-based Goal Model (VGM) and a Logical Role Model (LRM). The VGM captures the composition, temporal, and dependency relationships among different tasks/subtasks in a complex mission, as well as their relative importance to the overall mission. The LRM, on the other hand, is used to capture the physical or cyber functions required to achieve a particular goal (or successfully carry out a task). Based on this comprehensive LMM, users can feasibly model a complex mission, identify the criticality of each task/subtask, and evaluate the cyber resilience during the mission planning phase.



Attribute	Example Value
Node ID	Task 2013-10-3
Task Description	Have a teleconference with client A
Node Level	Root/Leaf/Intermediate Node
Node Type	Composition/Conjunctive/Disjunctive Node
Target Value	50
Priority/Weight	0.2
Accomplishment Status	Not yet start/In progress/Completed/Terminated
Progress Status	70%
Triggered by	Task 2013-10-1
Precedes	Task 2013-10-5
Parent Node	Task 2013-10
Child Node	Task 2013-10-3-1, Task 2013-10-3-2, Task 2013-10-3-3
Start Time	October 1, 2013 at 8:30:00 AM EDT
End Time	October 1, 2013 at 10:30:00 AM EDT

Fig. 3: VGM Node Attributes

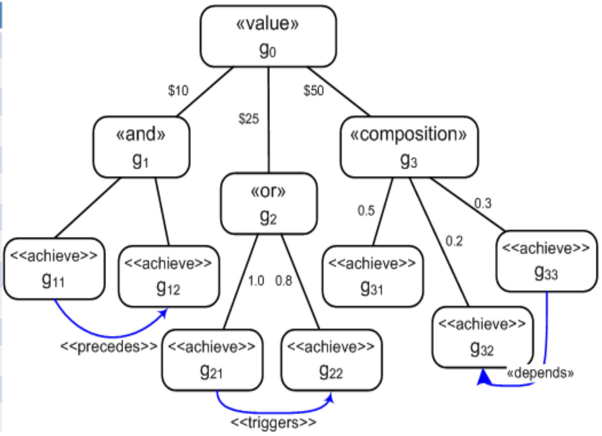


Fig. 4: An Example of VGM

Value-based Goal Model: Each node in VGM represents a task or goal that has to be achieved or maintained to ensure that the entire mission is accomplished. A higher level task (or goal) is represented as the parent node of multiple lower level subtasks (or sub-goals). Each node has a number of attributes to represent its current status as shown in Fig. 3. For example, each task is associated with a pre-assigned *Target Value* that represents its contribution to the overall accomplishment of its parent node, and the *Priority/Weight* attribute indicates the relative importance (criticality) of this node to its parent node. In our model, two other important attributes are *Accomplishment Status* and *Progress Status*. During the mission execution phase, these two attributes are periodically measured to evaluate the progress status of a mission task.

In our initial study, we identified three main entities for our VGM model: *goals*, *events*, and *parameters*. Specifically, a *goal* is an observable desired state of a mission/task, while an *event* is an observable phenomenon that occurs during the execution. *Parameters* of a goal or event provide the detailed information about the goal or the specific event. In our VGM, a complex mission is first decomposed into a set of simplified explicit tasks and the corresponding sub-tasks, and then represented by a hierarchical goal tree.

As illustrated in Fig. 4, upper level goals (*parent nodes*) can be decomposed into (also need to be supported by) a number of lower level sub-goals (*child nodes*). Each node (i.e., goal) has a pre-assigned value to represent its contribution to the overall mission. In addition, each parent goal's accomplishment relies on the accomplishment of its child nodes' goals, following the rules specified by mission commanders or Subject Matter Experts (SMEs). In our initial study, the achievement conditions for a parent node include *conjunctive*, *disjunctive*, and *composition* conditions. As shown in Fig. 4, the achievement condition and the value of a goal are represented via «and», «or», «composition», and «value» decorations of a node respectively.

As a starting point, we initially focused on modeling three temporal relationships between goals in the VGM, including *precedes*, *triggers*, and *subgoal* relationship. According to the ORD-Horn subclass defined in (Nebel B., et al., 1995), the formal definitions and appropriate timing constraints of these three temporal relationships are listed in Table 4.

Table 4: Temporal Relationships between Goals

Condition	Informal Constraint	Formal Constraint in ORD-Horn
$(\boldsymbol{a}, \boldsymbol{b}) \in \boldsymbol{precedes}$	*a* must be achieved before *b* can begin	$(a^+ \leq b^-) \wedge (a^+ \neq b^-)$
$(\boldsymbol{a}, \boldsymbol{b}) \in \boldsymbol{triggers}$	*a* must start before *b*; *b* must begin before *a* ends	$(a^- \leq b^-) \wedge (a^- \neq b^-) \wedge (b^- \leq a^+) \wedge (b^- \neq a^+)$
$(\boldsymbol{a}, \boldsymbol{b}) \in \boldsymbol{subgoal}$	*b* cannot start before *a* starts or end before *a* ends	$(a^- \leq b^-) \wedge (b^+ \leq a^+)$

Table 5 lists the various types and relationships between different goals and how to calculate their values in our VGM. Specifically, each node in VGM is a value goal. The *root goal*, *g*0, represents the overall value of a mission. The root value goal can be further decomposed into a set of *Composition*, *Conjunctive*, *Disjunctive goals* (as shown in Table 5), or *Leaf goals*. Each goal (i.e., node) has a pre-assigned “maxValue” to represent the expected value it can achieve if the corresponding task can be accomplished successfully. In our model, leaf goals have no subgoals. They directly contribute to the overall goal based on their parent’s type. Additionally, in VGM, only leaf goals are actively maintained by the system and need to be supported by the underlying cyber assets. As the leaf goal maintains (or fails), the overall value of a mission is aggregated based on the parent goals’ type, until the final goal is achieved (or aborted).

Logical Role Model: The LRM is designed to effectively capture corresponding cyber capabilities or functionalities required to achieve (or maintain) a particular task or goal. Working as an intermediate layer, our LRM maps the higher level logical mission elements onto the underlying network and cyber assets. By combining LRM with VGM, analysts will have a complete overview of the goals being pursued, the logical roles being performed to achieve those goals, and the corresponding network resources being used to carry out those roles. In our model, the logical dependency relationships are maintained at both mission planning and execution phases; not only for mission impact analysis, but also to improve the system’s resilience (e.g., alternative goals or redundant resources could be suggested or pre-assigned for critical tasks or mission elements, so that mission success can still be achieved even in the worst cases).

When modeling roles, the objective is to identify all the roles in the system as well as their interactions with each other. Given a valid VGM, we follow the following major steps to generate the corresponding LRM:

1) Create a role for each leaf-level goal in the goal model
2) If there are multiple ways to achieve a single goal, create a separate role for each approach and quantify the “goodness” of each approach (ranging from 0 to 1).
3) Identify information flows between the various roles

4) If two roles are tightly coupled, consider to combine them into a single role

5) Define the capabilities required to carry out each role

6) Determine the appropriate timing values associated with each role

Generally, to create a valid LRM, the first step is to create a single role for each leaf goal in the VGM. However, if we provide multiple ways to achieve a goal, the overall system resilience will increase. Documentation of the alternative approaches for each critical goal hence becomes very beneficial to mission assurance.

Once the roles have been identified, cyber capabilities required to carry out those roles can be further specified. In our model, cyber capabilities can be defined in terms of processing power, communication bandwidth, software and/or hardware specifications or requirements. The information flows between different roles can be used to implicitly define the communication capabilities for the logical roles. For example, if role A has to communicate with role B, the asset assigned for role A must be able to send/receive information to/from the asset assigned to role B. After assigning proper assets, specific communication and routing equipment can be further identified for the logical roles to provide the required communication capabilities.

Table 5: Goals Defined in VGM

Node Type	Definition	Value	Calculation
Value Goal	Each node in VGM is a value goal, and assigned with an associated value	Target Value	$maxValue(g) = \sum_{(g,g') \in subgoal} maxValue(g')$
		Current Value	$currentValue(g) = \sum_{(g,g') \in subgoal} currentValue(g')$
Composition Goal	Each subgoal contributes a percentage to its overall value, the total contributions must equal to 1	Target Value	$composition(g) \Rightarrow \left(\sum_{(g,g') \in subgoal} contribution(g') \right) = 1$
		Current Value	$currentValue(g) = maxValue(g) * \sum_{\substack{(g,g') \in subgoal \\ \wedge g' \in maintained}} contribution'(g')$
Conjunctive Goal	All subgoals have to be maintained; failure of any subgoal will reduce the parent's value to zero	Target Value	$conjunctive(g) \wedge (g, g') \in subgoal \Rightarrow maxValue(g') = maxValue(g)$
		Current Value	$currentValue(g) = maxValue(g) \times \prod_{(g,g') \in subgoal} \frac{currentValue(g')}{maxValue(g)}$
Disjunctive Goal	Value maintained if any subgoal is maintained,	Target Value	$disjunctive(g) \wedge (g, g') \in subgoal \Rightarrow maxValue(g') = maxValue(g) * contribution(g')$

	each subgoal has an associated contribution value	Current Value	$disjunctive(g) \Rightarrow currentValue(g) = \max(\{currentValue(g')\|(g, g') \in subgoal\})$

Note that to maintain and update information about currently available capabilities for supporting logical roles, real-time network monitoring and asset criticality analysis are required. In our framework, a cyber capability model (CCM) is designed to maintain the available capabilities of each cyber asset in a network, such as the current status (e.g., available, occupied, reserved), asset value, and dependency relationships. Other important information that should be maintained in the CCM includes host dependency, service map, and network topology. This knowledge can be directly derived by parsing the outputs of network monitoring and protocol analysis tools, such as Nmap (http://nmap.org/) and Wireshark (http://www.wireshark.org/), or leveraging state-of-the-art automated service discovery mechanisms developed by Tu et al. (2009) and Natarajan et al. (2012) into our framework.

4.2 Cyber Impact Analysis on Mission

After deriving the complete mission-to-asset dependency relationships via our logical mission models, the next step is to evaluate the potential impact of the lower level cyber incidents on the higher level mission elements. Following the same analysis method proposed by Jakobson (2011), the mission impact assessment process includes three major steps: (i) direct impact analysis of cyber incidents, (ii) cyber impact propagation analysis, and (iii) impact assessment on high level mission elements.

4.2.1 Direct Impact of Cyber Incidents

The *direct impact* can be defined as the loss of the *Asset Capacity* (*AC*) of an asset that is a direct target of an attack. As an internal feature of an asset, *AC* stays unchanged for the asset until its value is further reduced by another direct attack, or adjusted by external (human) operations (e.g., network operators may reset *AC* to *1* by recovering the damaged system). In our basic model, only software assets can be targets of direct attacks, and the initial value of *AC* is *1* (i.e., we assume that each asset is fully operational before it was attacked).

Particularly, if asset A does not depend on any other assets, then after it was directly attacked by attack X, its asset capacity can be expressed as follows:

$$AC_A(t^*) = Max[\, AC_A(t) - EP_A(t^*) \times IF_X(t^*), \quad 0] \qquad (7)$$

In Eq. 7, $AC_A(t)$ is the capacity of asset A at time t, $EP_A(t^*)$ is the exploit probability of the corresponding vulnerability on asset A at time t^*, $IF_X(t^*)$ is the impact factor of attack X at time t^*, and $AC_A(t^*)$ is the remained capacity of asset A at time t^*, given $t^* > t$.

Note that in a network environment, an asset could also be affected by the other assets it depends on. In this case, its AC will be determined by the combined effect of the other assets and the direct attack on it. For instance, if asset A depends on asset B and was a direct target of attack X, after being attacked its asset capacity should be:

$$AC_A(t^*) = Min[Max[\ AC_A(t) - EP_A(t^*) \times IF_X(t^*), \quad 0], \quad AC_B(t^*)] \qquad (8)$$

In Eq. 8, $AC_A(t)$ is the capacity of asset A at time t, $EP_A(t^*)$ is the exploit probability of the corresponding vulnerability on asset A at time t^*, $IF_X(t^*)$ is the impact factor of attack X at time t^*, $AC_B(t^*)$ is the capacity of asset B at time t^*, and $AC_A(t^*)$ is the remained capacity of asset A at time t^*, given $t^* > t$.

4.2.2 Propagation of Cyber Impact

In order to calculate the propagation of a direct impact through a network via the derived dependency relationships, we follow the same analysis method proposed by Jakobson (2011) and consider each asset as a generic node in a dependency graph, along with two kinds of specific "*AND*" and "*OR*" nodes to represent the logical dependency relationships between different elements. In this propagation model, the "*AND*" node defines that a parent node needs to depend on all of its children nodes, while the "*OR*" node defines that a parent node depends on the presence of at least one child node. Note that the "*OR*" dependency in our model is introduced to achieve better resilience, by providing redundant system, alternative functionality or performance to support a critical mission, task or operation. During the propagation of an attack, the capacities of all generic nodes in the attack path could be affected, either by a direct attack on it, or from a compromised child node it depends on.

To characterize the operational quality of each component or element at different levels in a mission-to-asset dependency graph, we further introduce the *Operational Capacity* (*OC*) as a universal measure in our model. The *Asset Capacity* (*AC*) presented previously is a specific form of the operational capacity provided by cyber assets. Similar to *AC*, *OC* is also measured in an interval [0, 1]. It indicates how much operational capacity that a cyber asset, service, task, or mission element can still provide after it was compromised or affected by an attack (directly or indirectly). Value 0 means that a component was totally destroyed (e.g., not operational), while value 1 means that it is still fully operational.

In our basic propagation model, the operational capacities of the "*AND*" and "*OR*" nodes are calculated as follows:

$$OC_{OR}(t) = \omega_i * OC_i(t) \mid \omega_1 * OC_1(t),\ \omega_2 * OC_2(t), \dots, \omega_n * OC_n(t), (1 \le i \le n) \qquad (9)$$

$$OC_{AND}(t) = Min\big(\omega_1 * OC_1(t),\ \omega_2 * OC_2(t), \dots, \omega_n * OC_n(t)\big), (1 \le i \le n) \qquad (10)$$

In Eq. 9 and Eq. 10, $OC_{OR}(t)$ is the operational capacity for an "*OR*" node at time *t*, $OC_{AND}(t)$ is the operational capacity for an "*AND*" node at time *t*. $OC_1(t)$, $OC_2(t)$, …, $OC_n(t)$ are operational capacities of the child nodes for the intended "*OR*" or "*AND*" nodes. ω_i is the pre-defined weight for each child node, based on its criticality to the parent node. Recursively applying Eq. 9 and Eq. 10 for all the nodes involved in the attack path, analysts can identify not only which asset could be affected, but also how much capacity will be lost due to the attack.

4.2.3 Impact Assessment on High Level Mission Elements

According to Jakobson (2011), during the mission execution stage, the real-time mission impact assessment depends on two major factors: (i) the impact that can be caused by the attacks, and (ii) in which state (e.g., planned, ongoing, or completed) of a mission or task.

For example, suppose that an attack *X* happened at time t^* (as shown in Fig. 5), and it could impact assets and services that support Tasks *A* through *E*. If those tasks have already been completed at time t^*, then those impacts should be irrelevant to the intended mission. If Task *F* is currently being executed, it can be affected if it relies on assets or services that can be impacted by attack *X*. Obviously, any other planned tasks that have not started yet but will depend on assets and services that could be impacted by attack *X* will probably be affected if no further countermeasures were taken.

Note that the planned tasks, such as Task *G* in Fig. 5, need to be analyzed carefully. As they have not yet been undertaken, their *OC* will not be accounted in the calculation of the overall *OC* of the intended mission. However, based on the (planned) mission asset mapping during the mission planning stage, we can calculate the potential impacts on those mission tasks, which could happen if we stick to the original asset mapping and network/system configurations. One advantage of our approach is that based on this real-time mission impact analysis, we can either reconfigure the corresponding network and systems, or replace a planned task with an alternative task to prevent or avoid the coming impacts and ensure a mission's success.

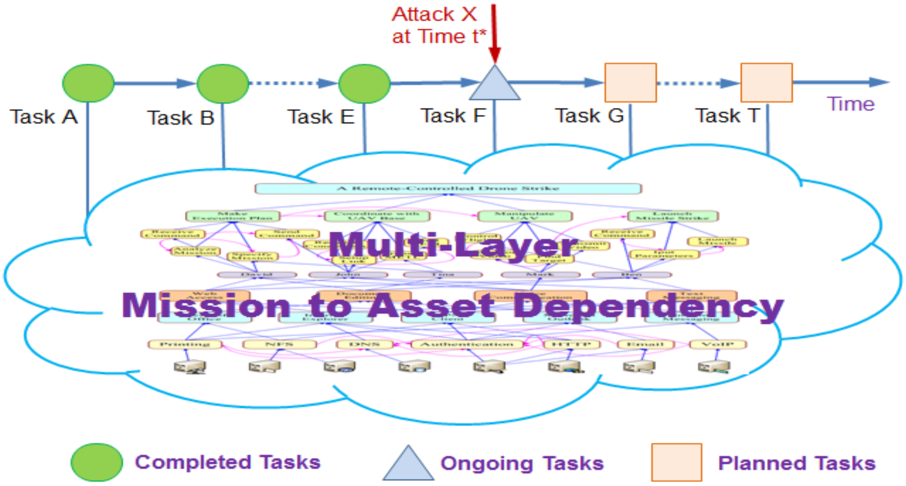


Fig. 5: Temporal Relations between Mission Tasks

In this mission impact analysis model, the execution of a mission is a process that unfolds step-by-step as time progresses. The initial operational capacity value of a mission or task is set as *OC=1*. This value could be steadily decreasing depending on the operational capacities of its executed stages and whether the corresponding assets and services were impacted by cyber attacks.

The calculation of the overall operational capacity of a mission will be calculated using Eqs. 7 - 10 accordingly for each potential attack path in our mission asset map, considering both dependency and temporal relationships. To achieve mission resilience, in the mission planning stage, we need to evaluate and compare different mission asset mapping and network configurations. For each mission asset mapping and network configuration, we calculate the operational capacity for both overall mission and the critical tasks. In this manner, we can find the best mapping and configuration to achieve the optimum value. In addition, to achieve better mission resilience, we can intentionally allocate/reserve redundant resources for critical leaf tasks and make critical task nodes as “*OR*” nodes (by adding alternative or backup tasks).

5 Asset Criticality Analysis and Prioritization

To identify the most critical cyber assets in supporting a critical task or operation, an effective measurement method is required for asset criticality ranking and prioritization. In our initial study, we prioritize asset criticality based on the cyber impact, mission relevance, and asset value analysis. In particular, the cyber impact and mission relevance can be evaluated by our attack risk prediction and impact propagation models described in Section 3 and Section 4. The asset value, in general, can be estimated by experienced network administrators, based on the amount of IT spending and the depreciation or amortization value of the assets (hardware and software).

Various decision making methods can be applied in our framework for mission asset criticality analysis and prioritization. As a starting point, we selected the standard Analytic Hierarchy Process (AHP) (Navneet B., et al., 2004) and Decision Matrix Analysis (DMA) (Shafter, 1976) methods in our initial study. Both of them can effectively prevent subjective judgment errors to increase the reliability and consistence of our analysis results.

5.1 AHP based Criticality Analysis

We first used AHP and pair-wise comparison matrix to calculate the relative value and importance of each mission related cyber asset. The general procedure for asset criticality analysis includes the following steps:

1) Modeling the problem as a hierarchy containing the decision goal, the alternatives for reaching it, and the criteria for evaluating the alternatives.
2) Establishing priorities among the elements of the hierarchy by making a series of judgments based on pair-wise comparisons of the elements. For example, when comparing asset value, network administrators might prefer database server over web server, and web server over desktop.
3) Synthesizing these judgments to yield a set of overall priorities for the hierarchy. This would combine network administrators’ judgments about different factors (such as asset value, potential loss, attack risk, and vulnerability severity) for different alternatives (e.g., Desktop A, Router H, Database P, etc.) into overall priorities for each asset.

4) Checking the consistency of the judgments.

5) Coming to a final decision based on the results of this process.

Fig. 6 shows a simple example of this process, in which three assets (i.e., desktop A, Router H and Database P) need to be prioritized based on three factors: *mission relevance*, *cyber impact* and *asset value*. In this example, we assume that cyber impact and mission relevance are both two times as important as asset value, and use a pair-wise comparison matrix to decide the proper weights for each factor.

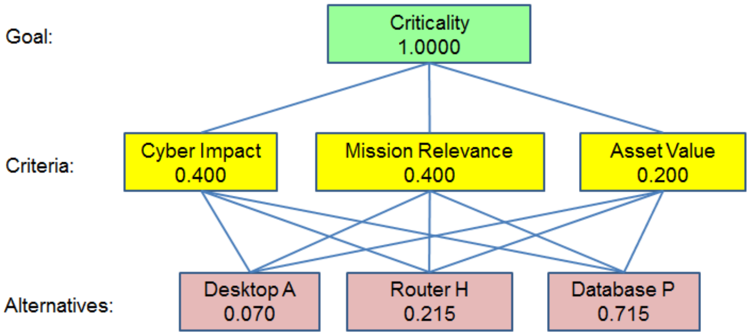


Fig. 6: Prioritization of cyber assets with AHP

As illustrated in Fig. 6, the weights of cyber impact and mission relevance are both 0.4, and the weight of asset value is 0.2. Additionally, each asset has a value vector to specify its relative value corresponding to the three factors, which will be used to calculate the asset's criticality (priority) based on the three weighted factors. Fig. 6 shows the prioritizing result of the three assets, in which Database P was the preferred entity, with a priority of 0.715. It was ten times as strong as Desktop A, whose priority was 0.07. Router H fell somewhere in between. Therefore, Database P is the most critical asset in this example, and it has to be well-protected from potential cyber attacks to assure mission success.

5.2 Grid Analysis based Prioritization

Grid analysis, also known as Decision Matrix Analysis, is another useful technique for making a decision among several options while taking many different factors into account. As the simplest form of Multiple Criteria Decision Analysis (MCDA) (http://en.wikipedia.org/wiki/Multi-criteria_decision_analysis), grid analysis is particularly powerful where users have a number of good alternatives to choose from and many different factors to take into account. To use grid analysis technique for decision making, first we need to list all the available options (*alternatives*) as rows on a table, and the factors (*criteria*) need to be considered as columns in the table. Then, we score each option/factor combination, weight the score, and add these scores up to give an overall score for each option in the table.

The step-by-step process of grid analysis technique can be illustrated as follows:

1) List all of the available options (*alternatives*) as the row labels on a table, and list the factors (*criteria*) as the column headings in the table.
2) Specify the relative importance of each factor, ranging from 0 (absolutely unimportant) to 5 (extremely important).
3) For each column, score each option/factor combination from 0 (poor) to 5 (very good), based on how well it possesses the corresponding factor.
4) Then, multiply each score from step 3) by the relative importance derived from step 2). This will give users weighted scores for each option/factor combination.
5) Finally, add up the corresponding weighted scores for each option. Options with higher scores are more important than the options with lower scores.

In our study, we initially considered the following factors to help security analysts decide which cyber asset or network service is more important than others:

- **Asset Value**: How important are the files and data stored in a host or server?
- **Cyber Severity**: What is the severity of a vulnerable service? This value can be derived from the CVSS score.
- **Mission/Task Dependency**: How important is the cyber asset or network service regarding to a critical mission and/or task?
- **Vulnerable Descendants**: How many descendants of this host could be potentially affected in the near future?

Additionally, the weight of each factor and the score of each option/factor combination are specified by the following rules:

- Based on its relative importance, each option service for each factor is scored from 0 to 5.
- The weight of each factor is normalized from 0 (not important) to 5 (extremely important).

Table 6 shows a simple example of grid analysis, in which a number of cyber assets and network services are listed. Specific weights have been assigned for four factors (*Asset Value*, *Cyber Severity*, *Mission/Task Dependency*, and *Vulnerable Descendants*). Each option/factor combination is assigned a particular value based on its relative importance decided by security analysts or domain experts.

Table 6: Grid Analysis for Mission Asset Prioritization

Factor	IP	User	Vulnerable	Asset	Cyber	Mission	Vulnerable	Total
Host	Address		Services	Value	Severity	Dependency	Descendants	Score
Weight				3	1	5	5	
Desktop_B	128.105.120.8	Jack	LICQ	1	2	5	3	45
AppServer_1	128.105.120.4	Mike	WebSphere	3	2	2	4	41
DBServer_1	128.105.120.5	John	Oracle DBMS	4	2	0	5	39
Desktop_C	128.105.120.14	Bob	Sysmgr GUI	2	0	1	2	21

Desktop_F	128.105.120.17	Mark	DCOM	2	0	2	1	21
Desktop_O	128.105.120.18	Bill	MySQL 5.1.x	2	0	0	0	6

The total score for each option is calculated and listed in the last column of Table 6. The "Desktop_B" (which is currently running "LICQ" service) has the highest score, which means it is the most important asset in supporting an intended mission. To protect "Desktop_B" from potential attacks, sufficient security resources or countermeasures should be applied. For instance, network administrators may shut down the vulnerable "LICQ" service to prevent the potential attacks. Note that we can virtually "shut down" a vulnerable service to demonstrate the corresponding consequences on the high level mission elements based on our logical mission models. If there is no big impact on the intended mission, or we can mitigate impact by reallocating alternative resource or goals, cyber resilience can be achieved to ensure mission assurance.

6 Future Work

Further investigation and research are still required, especially in the flowing fields:

- Efficient analytical models for mission-to-asset mapping (e.g., how to decompose a complex mission into a set of explicit tasks, identify mission-to-asset dependency, and allocate reliable cyber assets for critical tasks or mission elements.)
- Accurate network vulnerability and attack risk analysis models (e.g., how to configure/reconfigure a network to reduce aggregated network vulnerabilities; how to quickly detect and/or predict attack and attack path.)
- Practical mission impact assessment models (e.g., how to accurately model the direct impact of a cyber incident on a mission element; how to calculate the effect of a compromised cyber asset or failed mission element on the accomplishment of other mission elements.)
- Multi-layer graphical models (or a common operational picture) to effectively represent and display various inter- and intra- dependency relationships between different elements and components involved in CSA assessment
- Simple but meaningful metrics and corresponding evaluation algorithms or mechanisms for specific or general network security analysis

Note that the achievement of CSA rests in the ability to judiciously balance the above capabilities to handle the complexities of defensive operations. An integrated framework or software tool that leverages well-defined and developed technologies can significantly improve CSA and network security modeling, analysis, measurement, and visualization capabilities for security and mission analysts in enterprise network environments.

7 Summary

Meaningful security metrics are necessary to quantitatively evaluate and measure the operational effectiveness and system performance of a network. This chapter discussed how to effectively identify good metrics and evaluation methods for enterprise network situational awareness (SA) quantification and measurement. Metrics are tools that are designed to facilitate decision-making and improve performance and accountability through collection, analysis, and reporting of relevant performance-related data. Security measurement for CSA needs to carefully consider two distinct possible relationships: (i) How to define and use metrics as quantitative characteristics to represent the security state of a computer system or network, and (ii) How to define and use metrics to measure CSA from a defender's point of view. The multivariate nature of SA significantly complicates its quantification and measurement. State-of-the-art technologies provide useful descriptive information on security analysis, mission modeling, and situation management. The Common Vulnerability Scoring System has been widely adopted as the primary method for assessing the severity of computer system security vulnerabilities. The National Vulnerability Database provides CVSS scores for almost all known vulnerabilities. To evaluate the general security of an enterprise network based on vulnerability assessment, three security metrics are proposed: the vulnerable host percentage (VHP), CVSS severity score, and compromised host percentage (CHP). Attack graph based metrics can also be defined for network-level vulnerability assessment, such as the Number of Attack Paths, the Average Length of Attack Paths, and the Shortest Attack Path. Useful metrics can also be based on modeling (i) the logical relations that allow us to model the propagation of the impacts through the network, and (ii) the computational relations that allow us to calculate the level of those impacts. Analysts can feasibly model a complex mission, identify the criticality of each task/subtask, and evaluate the cyber resilience during the mission planning phase. After deriving the complete mission-to-asset dependency relationships via the logical mission models discussed in this chapter, the next step is to evaluate the potential impact of the lower level cyber incidents on the higher level mission elements. Using the real-time mission impact analysis, network operators can either reconfigure the corresponding network and systems, or replace a planned task with an alternative task to prevent or avoid the coming impacts and ensure a mission's success. AHP and pair-wise comparison matrix can help calculate the relative value and importance of each mission related cyber asset. Effectively identifying the right metrics to measure security preparedness and awareness within an organization is a hard and complicated problem. To be valuable, security metrics must be meaningful to organizational goals or key performance indicators. Security analysts should review their specific metrics currently in place and ensure that they are aligned with the overall industry standards and their particular organizational and business goals.

Disclaimer

This paper is not subject to copyright in the United States. Commercial products are identified in order to adequately specify certain procedures. In no case does such identification imply recommendation or endorsement by the National Institute of Standards and Technology, nor does it imply that the identified products are necessarily the best available for the purpose.

References

Alberts C., et al. (2005). *Mission Assurance Analysis Protocol (MAAP): Assessing Risk in Complex Environments. CMU/SEI-2005-TN-032.* Pittsburgh, PA: Carnegie Mellon University.

Ammann P., et al. (2002). Scalable, Graph-based Network Vulnerability Analysis. *the 9th ACM Conference on Computer and Communications Security.*

Bolstad C. and Cuevas H. (2010). Integrating Situation Awareness Assessment into Test and Evaluation. *The International Test and Evaluation Association (ITEA)*, 31: 240–246.

Cheung S., et al. (2003). Modeling Multi-Step Cyber Attacks for Scenario Recognition. *the 3rd DARPA Information Survivability Conference and Exhibition.* Washington D. C.

Dahl, O. (2005). *Using colored petri nets in penetration testing. Master's thesis.* Gjøvik, Norway: Gjøvik University College.

Durso F., et al. (1995). Expertise and chess: A pilot study comparing situation awareness methodologies. *In experimental analysis and measurement of situation awareness*, (pp. 295–303).

Endsley M. R. and Garland D. J. (eds). (2000). *Theoretical underpinnings of situation awareness: A critical review. In Situation awareness analysis and measurement.* Mahwah, NJ: LEA.

Endsley, M. R. (1988). Situation awareness global assessment technique (SAGAT). *the National Aerospace and Electronics Conference (NAECON).*

Endsley, M. R. (1990). Predictive utility of an objective measure of situation awareness. *the Human Factors Society 34th Annual Meeting*, (pp. 41–45).

Endsley, M. R. (1995). Measurement of situation awareness in dynamic systems. *Human Factors*, 37(1), 65–84.

Endsley, M. R., et al. (1998). A comparative evaluation of SAGAT and SART for evaluations of situation awareness. *the Human Factors and Ergonomics Society Annual Meeting*, (pp. 82–86).

Fracker, M. (1991a). Measures of situation awareness: Review and future directions (Report No. AL-TR-1991-0128). Wright-Patterson Air Force Base, OH: Armstrong Laboratories.

Fracker, M. (1991b). Measures of situation awareness: An experimental evaluation (Report No. AL-TR-1991-0127). Wright-Patterson Air Force Base, OH: Armstrong Laboratories.

Gomez M., et al. (2008). *An Ontology-Centric Approach to Sensor-Mission Assignment.* Springer.

Goodall J., et al. (2009). Camus: Automatically Mapping Cyber Assets to Missions and Users. *IEEE Military Communications Conference*. Boston MA.

Grimaila M., et al. (2008). Improving the Cyber Incident Mission Impact Assessment Processes. *the 4th Annual Workshop on Cyber Security and Information Intelligence Research.*

Grimaila M., et al. (2009). Design Considerations for a Cyber Incident Mission Impact Assessment (CIMIA) Process. *the 2009 International Conference on Security and Management (SAM09).* Las Vegas, Nevada.

Harwood K., et al. (1988). Situational awareness: A conceptual and methodological framework. *the 11th Biennial Psychology in the Department of Defense Symposium*, (pp. pp. 23–27).

Hecker, A. (2008). On System Security Metrics and the Definition Approaches. *the 2nd International Conference on Emerging Security Information, Systems and Technologies.*

Heyman T., et al. (2008). Using security patterns to combine security metrics. *the 3rd International Conference on Availability, Reliability and Security*.

Holsopple J., et al. (2006). TANDI: Threat Assessment of Network Data and Information. *SPIE Defense and Security Symposium.*

Holsopple J., et al. (2008). FuSIA: Future Situation and Impact Awareness. *Information Fusion*.

http://cve.mitre.org/cve/. (n.d.). Retrieved May 8, 2014

http://en.wikipedia.org/wiki/Multi-criteria_decision_analysis. (n.d.). Retrieved May 8, 2014

http://nmap.org/. (n.d.). Retrieved May 8, 2014

http://nvd.nist.gov/. (n.d.). Retrieved May 8, 2014

http://www.first.org/cvss/. (n.d.). Retrieved May 8, 2014

http://www.first.org/cvss/cvss-guide. (n.d.). Retrieved May 8, 2014

http://www.microsoft.com/en-us/download/details.aspx?id=7558. (n.d.). Retrieved May 8, 2014

http://www.mindtools.com/pages/article/newTED_03.htm. (n.d.). Retrieved May 8, 2014

http://www.openvas.org/. (n.d.). Retrieved May 8, 2014

http://www.tenable.com/products/nessus. (n.d.). Retrieved May 8, 2014 *http://www.wireshark.org/*. (n.d.). Retrieved May 8, 2014

Jakobson G. (2011). Mission Cyber Security Situation Assessment Using Impact Dependency Graphs. *the 14th International Conference on Information Fusion (FUSION)* (pp. 1 - 8). Chicago, IL: IEEE.

Jansen, W. (2009). *Directions in Security Metrics Research.* National Institute of Standards and Technology, Computer Security Division.

Jones D. and Endsley M. R. (2000). Examining the validity of real-time probes as a metric of situation awareness. *the 14th Triennial Congress of the International Ergonomics Association.*

King, S. (2012). *Defense Cyber S&T Strategy*. Retrieved from http://www.security-innovation.org/ pdfs/King_final.pdf

Kotenko I., et al. (2006). Attack graph based evaluation of network security. *the 10th IFIP TC-6 TC-11 international conference on Communications and Multimedia Security,* (pp. 216-227).

Lewis L., et al. (2008). Enabling Cyber Situation Awareness, Impact Assessment, and Situation Projection. *Situation Management (SIMA).*

Lindstrom, P. (2005). Security: Measuring Up. Retrieved from http://searchsecurity.techtarget.com/tip/Security-Measuring-Up

Manadhata P. and Wing J. (2011). An Attack Surface Metric. *Software Engineering, IEEE Transactions on, vol.37, no.3,* 371-386.

Matthews M., et al. (2000). Measures of infantry situation awareness for a virtual MOUT environment. *the Human Performance, Situation Awareness and Automation: User-Centered Design for the New Millennium.*

McDermott, J. (2000). Attack net penetration testing. *Workshop on New Security Paradigms.*

Meland P. and Jensen J. (2008). Secure Software Design in Practice. *the 3rd International Conference on Availability, Reliability and Security.*

Musman S., et al. (2010). *Evaluating the Impact of Cyber Attacks on Missions.* MITRE Technical Paper #09-4577.

Natarajan A., et al. (2012). NSDMiner: Automated discovery of network service dependencies. *INFOCOM* (pp. 2507-2515). IEEE.

Navneet B., et al. (2004). *Strategic Decision Making: Applying the Analytic Hierarchy Process.* London: Spring-Verlag.

Nebel B., et al. (1995). Reasoning about temporal relations: a maximal tractable subclass of Allen's interval algebra. *Journal of the ACM (JACM)*, vol. 42, no. 1, pp. 43-66.

Noel S., et al. (2004). Correlating Intrusion Events and Building Attack Scenarios through Attack Graph Distance. *the 20th Annual Computer Security Conference.* Tucson, Arizona.

Ou X., et al. (2006). A Scalable Approach to Attack Graph Generation. *the 13th ACM Conference on Computer and Communication Security (CCS)*, (pp. 336-345).

Qin X. and Lee W. (2004). Attack Plan Recognition and prediction Using Causal Networks. *the 20th Annual Computer Security Applications Conference.*

Salerno J., et al. (2005). A Situation Awareness Model Applied to Multiple Domains. *Multisensor, Multisource Information Fusion*.

Salerno, J. (2008). Measuring situation assessment performance through the activities of interest score. *the 11th International Conference on Information Fusion.*

Shafter, G. (1976). *Mathematical Theory of Evidence.* Princeton University Press.

Sheyner O., et al. (2002). Automated Generation and Analysis of Attack Graphs. *the 2002 IEEE Symposium on Security and Privacy*, (pp. 254-265).

Singhal A., et al. (2010). Ontologies for modeling enterprise level security metrics. *the 6th Annual Workshop on Cyber Security and Information Intelligence Research.* ACM.

Strater L., et al. (2001). *Measures of platoon leader situation awareness in virtual decision making exercises (No. Research Report 1770).* Army Research Institute.

Tadda G., et al. (2006). Realizing Situation Awareness within a Cyber Environment. *Multisensor, Multisource Information Fusion: Architectures, Algorithms, and Applications* (p. 1−8). Orlando: SPIE Vol.6242.

Taylor, R. (1989). Situational awareness rating technique (SART): The development of a tool for aircrew systems design. *the AGARD AMP Symposium on Situational Awareness in Aerospace Operations, CP478.*

Tu W., et. al. (2009). Automated Service Discovery for Enterprise Network Management. Stony Brook University. Retrieved May 8, 2014, from http://www.cs.sunysb.edu/~live3/research/asd_ppt.pdf

Vidulich M. (2000). Testing the sensitivity of situation awareness metrics in interface evaluations. *Situation awareness analysis and measurement*, 227–246.

Wang J., et al. (2009). Security Metrics for Software Systems. *the 47th Annual Southeast Regional Conference.*

Watters J., et al. (2009). *The Risk-to-Mission Assessment Process (RiskMAP): A Sensitivity Analysis and an Extension to Treat Confidentiality Issues.*

Zhou S., et al. (2003). Colored petri net based attack modeling. *Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing: the 9th International Conference* (pp. vol. 2639, pp. 715–718). Chongqing, China: Springer.