STRAM: Measuring the Trustworthiness of Computer-based Systems [1]

A

JIN-HEE CHO, Virginia Tech, USA
SHOUHUAI XU, The University of Texas at San Antonio, USA
PATRICK M. HURLEY, US Air Force Research Laboratory
MATTHEW MACKAY, UK Defence Science and Technology Laboratory
TREVOR BENJAMIN, UK Defence Science and Technology Laboratory
MARK BEAUMONT, Defence Science and Technology Group, Australia

Various system metrics have been proposed for measuring the quality of computer-based systems, such as dependability and security metrics for estimating their performance and security characteristics. As computer-based systems grow in complexity with many sub-systems or components, measuring their quality in multiple dimensions is a challenging task. In this work, we tackle the problem of measuring the quality of computer-based systems based on the four key attributes of trustworthiness we developed, namely security, trust, resilience and agility. In addition to conducting a systematic survey on metrics, measurements, attributes of metrics and associated ontologies, we propose a system-level trustwothiness metric framework that accommodates four submetrics, called STRAM (Security, Trust, Resilience, and Agility Metrics). The proposed STRAM framework offers a hierarchical ontology structure where each submetric is defined as a sub-ontology. Moreover, this work proposes developing and incorporating metrics describing key assessment tools, including Vulnerability Assessment, Risk Assessment and Red Teaming, to provide additional evidence into the measurement and quality of trustworthy systems. We further discuss how assessment tools are related to measuring the quality of computer-based systems and the limitations of the state-of-the-art metrics and measurements. Finally, we suggest future research directions for system-level metrics research towards measuring fundamental attributes of the quality of computer-based systems and improving the current metric and measurement methodologies.

CCS Concepts:•**Security and privacy** → **Systems security;**

1. INTRODUCTION

1.1. Motivation

Measuring the quality of a computer-based system is critical to building a trustworthy system because such a measurement can be an objective indicator to validate the quality of the system with a certain level of confidence. In the past, security and dependability have been discussed as major system metrics to measure the quality of a computer-based system [Avizienis et al. 2004; Nicol et al. 2004; Pendleton et al. 2016]. However, they cannot adequately consider the multidimensional quality of computer-based systems, particularly associated with hardware, software, networks, human factors, and physical environments. To address this issue, we choose to measure trustworthiness to derive a holistic quality measurement of the system. Up to now, the concept of trustworthiness as a system metric has received very little attention based on the lack of literature from research conferences and journals and its research is still in its infant stage.

The Technical Cooperation Program (TTCP) [2] initiated this effort in cybersecurity under the banner of the Cyber Strategic Challenge Group (CSCG). In particular, under the CSCG, a *Trustworthy Systems Working Group* (TSWG) was formed in 2014 to share and conduct collaborative research with the following four key activities:

(1) Tools & Methods: Common and sharable tools to search for, find, and fix vulnerabilities as well as to design cyber-hardened systems;
(2) Building System Composition: Developing ways to build trustworthy systems from components of differing levels of trust, such as developing trustworthy design patterns;
(3) Review & Assessment: Developing assessment methodologies to include red teaming for the effective use of tools and techniques; and
(4) Metrics & Measurement: Develop meaningful and repeatable metrics to measure the trustworthiness of systems.

[1]Author's addresses: Jin-Hee Cho is with the Department of Computer Science at Virginia Tech, VA, USA. This work was done when Jin-Hee Cho was with US Army Research Laboratory, Adelphi, MD, USA. Shouhuai Xu is with the Department of Computer Science at The University of Texas at San Antonio, TX, USA. Pactrick M. Hurley is with the US Air Force Research Laboratory, NY, USA. Matthew Mackay and Trevor Benjamin are with the UK Defense Science and Technology Laboratory, Salisbury, UK. Mark Beaumont is with the Defence Science and Technology Group, Adelaide, Australia. Correspondence: Jin-Hee Cho (jicho@vt.edu)
[2]TTCP is an international organization aiming to collaborate and exchange defense scientific and technical research and information, harmonize and align defense research programs by sharing or exchanging research activities between the five nations, Australia, Canada, United Kingdom, United States, and New Zealand [TTCP 2014].

Through sharing and exchanging defense research and technology of each member nation and conducting collaborative research, TSWG aims to build a cyber-hardened system which is highly trusted, resilient and agile under highly dynamic system and threat conditions. In this paper, the members in the TSWG describe the development of metrics for measuring the multidimensional trustworthiness of computer-based systems.

In order to measure the trustworthiness of computer-based systems, we propose a metric framework called STRAM (Security, Trust, Resilience, and Agility Metrics). We developed these four key metrics with the goal of covering the multidimensional quality of computer-based systems based on fundamental attributes.

1.2. Key Contributions

This work makes the following contributions:

— ***Using the concept of trustworthiness as a system-level metric***. Although trustworthiness can be defined differently, we adopt the concept of trustworthiness as a system-level metric embracing the four key aspects of the quality of computer-based systems, namely security, trust, resilience, and agility.
— ***Systematic definition of resilience and agility metrics***. Although security and trust metrics have been extensively investigated in the literature, resilience and agility metrics are much less understood and investigated. This work presents a systematic definition of resilience and agility, making a significant step beyond what is currently available.
— ***Ontology-based metric framework***. This work proposes an ontology-based metric framework for measuring trustworthiness, consisting of security, trust, resilience, and agility submetrics. The proposed metric framework, called STRAM, takes a hierarchical ontology methodology where the entire trustworthiness metric ontology is composed of four submetric ontologies. We use an ontology tool, called Protégé [Stanford Center for Biomedical Informatics Research 2015], to visualize the hierarchical structure of ontologies in STRAM.
— ***Consideration of the diverse aspects of system components***. The proposed metric framework, STRAM, is designed to measure a system-level quality that embraces the diverse aspects of system components. STRAM focuses on measuring the quality of a system across hardware, software, networks, human factors, and physical environments while most existing metrics only aim to measure a particular component or aspect of a system.
— ***Investigating the relationships between metrics and assessment tools***. Various assessment tools, including vulnerability assessment, penetration testing, risk assessment and red teaming, have been used to evaluate the quality of a system. We also provide a set of metrics to evaluate the assessment tools and discuss how the assessment tools measure factors affecting the quality of a system, such as attack, defense mechanisms, vulnerabilities, and the validity of a metric framework.
— ***Discussion of limitations of existing approaches and future work suggestion***. We discuss the significant hurdles and open challenging questions in the system trustworthiness metric research domain. In addition, we suggest future research directions for system-level measurement of trustworthy systems.

A very preliminary version of this paper was published in [Cho et al. 2016]. The present work substantially extends [Cho et al. 2016] in that (1) a more in-depth survey was conducted for security and trust metrics, measurements, and metric ontologies; (2) systematic definitions of resilience and agility metrics are provided; (3) the ontology of each submetic (i.e., security, trust, resilience, and agility) is addressed with details of each attribute contributing to the submetric ontology; (4) metrics of assessment tools are discussed with specific metrics and measurement for each tool (i.e., vulnerability assessment, risk assessment, and red teaming); (5) the relationships between system vulnerability, attack, defense, assessment tools, and the validity of the proposed system metric framework, namely STRAM, are investigated; (6) the limitations of the existing system metrics are discussed; and (7) future research directions are discussed based on the identified gaps between the ideal metric and the existing state-of-the-art metrics.

Note that the scope of this work is to define and discuss the concept of trustworthiness as a system-level metric where we constructed a hierarchical, ontology-based system metric framework based on the four key submetrics, including security, trust, resilience, and agility. The focus of this work is to identify and discuss key metrics that should be considered to measure the trustworthiness quality of any computer-based systems. Depending on different characteristics or configurations of system platforms or environments, different ways of estimating particular metrics can be derived. For example, to measure agility of a defensive action, one system may use the mean time to take an adaptive action while the other system may estimate the time an attacker is removed after it penetrated into the system. That is, we do not include an in-depth discussion on how each system can use what kinds of metrics to measure a certain

metric attribute in this work as we limit our scope to propose a high-level system metric framework in this work. In addition, considering the large volume of this paper discussing the key contributions, we will leave the quantitative calculations of the four metrics for our future work. We discuss more detailed future work directions in Section 8.2. However, in order to give a sense on how each submetric is measured in practice, we show some example metrics as shown in Table III.

1.3. Comparison with Existing Surveys on System-Level Metrics

Some existing survey papers discussed system-level metrics. Avizienis et al. [2004] discussed dependability to indicate the overall quality metric of a system based on four subattributes including security, safety, reliability, and maintainability. Pendleton et al. [2017] proposed a system-level security metric framework based on the interactions between attackers and defenders in order to represent dynamic system security metrics. Ramos et al. [2017] mainly surveyed model-based network security metrics considering a suite of probabilistic and/or analytical models to conduct quantitative evaluation of network security.

This paper is different from the above survey papers [Avizienis et al. 2004; Pendleton et al. 2017; Ramos et al. 2017] in that the proposed STRAM framework aims to define a system-level metric that can capture the quality of system performance and security in terms of both static and dynamic nature. To achieve this goal, we consider four attributes of system quality which are security, trust, resilience, and agility. Trust can consider the effect of human factors on system quality in addition to other system factors while resilience and agility can reflect more dynamic nature of system quality which is closely related to enhancing security. We consider these four system quality aspects under the roof of the concept of trustworthiness. This is unique in that no prior work has considered based on a hierarchical, ontological metric framework that uses 'trustworthiness' to represent the overall system metric. The most comparable metric in the literature is the dependability metric proposed by Avizienis et al. [2004]. However, the dependability metric in [Avizienis et al. 2004] does not cover dynamic nature of system quality such as resilience and agility although maintainability, as one of dependability attributes, considers some part of resilience such as fault-tolerance and/or recovery. But in [Avizienis et al. 2004], no clear distinction between fault-tolerance and recovery (e.g., self-healing or self-recovery) or between trustworthiness and dependability is made although they should be considered differently.

1.4. Paper Organization

The rest of this paper includes the following sections:

— Section 2 gives background information to provide an understanding of the basic concepts of metrics and measurements. In addition, we discuss criteria for valid metrics, categories and properties of the metrics, existing metric attributes, metric scales, and methods of measurements.
— Section 3 describes the main components of computer-based systems, threats, and key attributes of trustworthiness in terms of security, trust, resilience, and agility. These four attributes are discussed in terms of their definitions.
— Section 4 surveys metric ontologies measuring computer-based system attributes, including ontologies of security, trust, resilience, and agility.
— Section 5 gives the detailed description of each submetric ontology in the STRAM framework.
— Section 6 discusses the key metrics of vulnerability assessment, risk assessment, and red teaming.
— Section 7 investigates how assessment tools and the metric frameworks are related to attacks, vulnerabilities, and defense mechanisms. Further, we discuss how they are related to each other and ultimately how they are related to a system-level metric that measures the trustworthiness of a computer-based system. In addition, we identify the limitations of the existing metric techniques.
— Section 8 summarizes the key contents covered in this paper and suggests future research directions.

2. METRICS AND MEASUREMENTS

This section discusses background knowledge to understand the proposed metric framework to measure the quality of systems. We briefly discuss the definitions of metrics and measurements and their basic classifications used in the literature.

2.1. Metrics

A "metric" is used to indicate "a precisely defined method which is used to associate an element of an (ordered) set V to a system S" [Böhme and Freiling 2008]. We can formalize a function of metric, M, where system S is mapped to elements of an ordered set V, as:

$$M : S \rightarrow V \tag{1}$$

2.1.1. **Criteria for Valid Metrics**. The validity of a metric has been discussed based on the following key criteria [Slayton 2015]:
— *Objectivity*: A metric should provide quantification providing mechanical objectivity based on a set of rules;
— *Efficiency*: Automated quantification provides high efficiency and accordingly increases system efficiency based on quantified metrics;
— *Control and feedback*: The measured metric can provide feedback or means of controlling decisions; and
— *Learning*: By measuring risk, a metric can improve quality of systems based on learning from the results.

2.1.2. **Categories of Metrics**. National Institute of Standards and Technology (NIST) [Information Technology Laboratory, Software and Systems Division 2016] classifies metrics and measures based on the following categories:
— *Primitive vs. Derived*: Primitive metrics are captured based on raw data of measurements (e.g., the number of lines of codes, the number of anomalous traffic flows), while derived metrics are obtained through a derivation process based on an aggregation function (e.g., a weighted sum to measure service reliability).
— *Static vs. Dynamic*: Static metrics only rely on measurements from a system state at a particular time, while dynamic metrics measure the system state adapting to dynamic changes across time.
— *Objective vs. Subjective*: Objective metrics use a set of certain rules and mostly follow the measurement process in an automated and repeated way. In contrast, subjective metrics may reflect judgments by human analysts, operators, or users while capturing the learning capability that may not be obtainable from simple objective but automated measurements.
— *Aspect Measured*: Different aspects of measurements can be used including (1) *size* (i.e., raw measurements such as the number of lines of code or the number of anomalous traffic flows); (2) *complexity* (i.e., structural, computational, algorithmic, logical, functional complexity); and (3) *quality* (i.e., performance attributes, as used in this paper - reliability, availability, usability, etc.).

2.1.3. **Properties of Metrics**. It is challenging to select a right metric to measure fundamental attributes of a system. Böhme and Freiling [2008] define the properties of a good metric in measuring an attribute of the system:
— *Relation*: When comparing the performance based on an attribute function, $c(\cdot)$, of two systems, denoted by x and y, a sensible relation should exist such that $c(x) > c(y)$ or $c(x) < c(y)$; and
— *Operation*: A meaningful operation should be applied in measuring attributes of a system based on observed system features (e.g., adding or multiplying the number of compromised nodes to measure network vulnerability).

2.1.4. **Existing Metric Attributes**. The quality of a system can be discussed using many attributes. For example, Avizienis et al. [2004] claim that the fundamental attributes of metrics reflecting the quality of a system are functionality, performance, dependability, coupled with security and cost. "Usability, manageability, and adaptability" are discussed as the factors affecting dependability and security. Moreover, they discuss security in terms of availability, integrity, and confidentiality, while using dependability to embrace reliability, availability, integrity, safety, and maintainability. As another example, Hasselbring and Reussner [2006] define the key attributes of software trustworthiness in terms of correctness, safety, quality of service (i.e., availability, reliability, performance), security (i.e., availability, integrity, confidentiality), and privacy. They suggest a holistic approach to capture the complexity of a system based on multidimensional optimization techniques.

As discussed above, although security and dependability are commonly mentioned as system-level metrics, *trustworthiness* has been explored in the context of socio-technical systems [Mohammadi et al. 2014] and cyber sensing [Xu 2010], which can often be observed in Internet-based software. They categorize trustworthiness attributes in terms of security, compatibility, configuration quality, compliance, cost, data quality, dependability, performance, usability, correctness, and complexity. There have been some studies on using provenance to evaluate trustworthy information [Dai et al. 2012; Xu et al. 2009, 2010]. However, the studies mentioned above [Avizienis et al. 2004; Dai et al. 2012; Hasselbring and Reussner 2006; Mohammadi et al. 2014; Xu 2010; Xu et al. 2009, 2010] do not provide a holistic measurement perspective of a system.

The question of "how to achieve trustworthiness" is closely related to finding the answer for what makes systems trustworthy. Avizienis et al. [2004] discuss the means to security and dependability in terms of fault prevention, fault tolerance, fault removal, and fault forecasting. Although they are useful for build-

ing secure and dependable systems, it is not clear how they are associated with the quality of procedures / tools for assessing the quality of systems.

Although the above works [Avizienis et al. 2004; Hasselbring and Reussner 2006; Mohammadi et al. 2014] discuss the major metrics, including key attributes to measure the quality of a system, they do not address: (1) how each attribute is related to other attributes; (2) how an attribute's meaning overlaps with that of the other attributes'; and (3) how attributes are hierarchically structured with a full-fledged granularity of sub-attributes representing the quality of multidimensional system domains. These questions are addressed in this paper.

Trustworthiness of systems is related to the following components: (1) a system of concern and its features, states, and behavior; (2) threats, including faults, errors, and failures caused by deliberate actions (i.e., attacks) or non-deliberate actions (e.g., mistakes by a user or system operator); (3) means to build trustworthy systems (e.g., system and security protocols or mechanisms); and (4) quality of assessment tools (e.g., red teaming, vulnerability assessment, penetration testing).

2.2. Measurements

As mentioned above, a metric indicates the quality of an object and is closely related to a measurement based on evidence. "To measure" means assigning an element of a scale to an object for quantifying an attribute of the object [Böhme and Freiling 2008]. A measurement uses an abstraction to reduce the complexity of representing multiple attributes of a system to a single symbol. By using the measurement based on a single symbol, we can classify and compare multiple systems based on the metrics quantifying their attributes [Böhme and Freiling 2008].

2.2.1. **Measurement Scales**. We collect the result of measurements from data which can be categorized as different *scales*. The scale refers to the range of V in Eq. (1) and addresses the relations between elements in V. The types of scales are classified as follows [Böhme and Freiling 2008]:

— *Nominal scale*: This is the simplest scale and is also called a categorical scale, where V is an unordered discrete set (e.g., yes/no, 0/1, male/female, red/blue/yellow).
— *Ordinal scale*: This uses an ordered discrete set for V which allows comparison of multiple systems using an attribute (e.g., less than or larger than). In this scale, monotonic mapping in the ordering relation is preserved such that if $a < b$ where $a, b \in S$, then $M(a) < M(b)$.
— *Interval scale*: The comparison of two systems can be measured based on the difference operator, such as adding, subtracting, multiplying, or dividing a constant, to measure the relative distance between two scale points.
— *Ratio scale*: This is an extension of an interval scale where the origin is naturally defined such as 0. It can measure length, mass, time period, or monetary value.

Nominal and ordinal scales belong to qualitative scales while interval and ratio scales are quantitative scales. Quantitative scales are more powerful by allowing the use of parametric statistics based on a distribution assumption in which inference is possible [Böhme and Freiling 2008].

2.2.2. **Measurement Methods**. In this work, we classify methods of measurements with two approaches, modeling and analysis-based measurements and experiments-based measurements.

Modeling and Analysis-based Measurements. These measurements aim to measure the quality of a system and have been explored using various types of analytical methods (e.g., stochastic models, stochastic petri networks, Markov process) [Cho 2015; Li et al. 2011; Xu 2014a; Xu et al. 2015], simulation [Han et al. 2014; Xu et al. 2015; Zheng et al. 2015], and emulation models [Chan et al. 2015].

In particular, Cybersecurity Dynamics approach has been proposed for modeling and analyzing cybersecurity as the methodology from a holistic perspective [Xu 2014a]. This approach can accommodate explicitly the dynamic threats, and therefore is suitable for modeling and analyzing not only security, but also potentially resilience and agility. Several variations of cybersecurity dynamics have been investigated, including preventive and reactive defense dynamics [Xu and Xu 2012; Xu et al. 2012a,b; Zheng et al. 2016], adaptive defense dynamics [Xu et al. 2014], proactive defense dynamics [Han et al. 2014], and active defense dynamics [Lu et al. 2013; Zheng et al. 2015; Xu et al. 2015]. Holistic security requirements in the context of socio-technical systems have been investigated in [Li et al. 2016]. Despite the prior studies mentioned above, there are no prior studies on holistic resilience and agility, which will be investigated in this paper.

Experiment-based Measurements. These approaches often use known measurement tools or conduct empirical studies to measure the quality of a system in terms of particular metric attributes.

Measurement using tools: The quality of assessment, testing, or verification (e.g., vulnerability assessment, penetration testing, risk assessment or red teaming) significantly affects the level of confidence based on the uncertainty introduced to the measurements by the tools [Jr. et al. 2003]. This is an important matter because uncertainty from unknown attacks, unknown vulnerabilities or unknown risk can often hinder appropriate actions to prevent, tolerate, remove, or forecast faults, which are the main causes of system errors or failures.

The four key attributes of trustworthiness are related to the degree of threat, uncertainty, asset importance, and risk appetite (e.g., risk-seeking, risk-neutral, risk-averse). Varying the severity of threats can modify the effectiveness of red teaming by tailoring their design and implementation to test a system's resilience against attacks or faults. Thus, varying the level of threats considered in red teaming affects the level of assessed trustworthiness in a highly dynamic, hostile environment.

We discuss the following tools which are well known to measure vulnerabilities of a system:

— *Vulnerability assessment* (VA) refers to the process for examining a system to identify its weaknesses that may provide an attack surface for adversarial entities to perform attacks [Goel and Mehtre 2015].
— *Penetration testing* (PT) is an evaluation / verification process that tests various features of operations / functionalities of a system for finding vulnerabilities exploitable by attackers [Antunes and Vieira 2009; Hayes 2016; RedTeams 2013].
— *Red teaming* (RT) is an assessment process for identifying vulnerabilities or weaknesses in various aspects of a system, aiming to improve the quality of a system throughout its development process and even during its use [Wood and Duggan 2000].

Although PT and VA overlap in identifying vulnerabilities and PT is sometimes even interchangeably used with VA, PT is a more specific, goal-oriented testing process, whereas VA provides a list of vulnerabilities of a system as well as their priorities to be fixed. In particular, PT has a clear goal of determining the exploitability of identified vulnerabilities based on the already performed VA (e.g., an unauthorized user tries to gain access to a system by penetrating system security and defense mechanisms) [Goel and Mehtre 2015]. On the other hand, RT is more encompassing than PT because RT is designed to enhance security by identifying vulnerabilities and improving defense strategies (e.g., countermeasures against attacks or prevention mechanisms for vulnerabilities) [Wood and Duggan 2000].

Empirical measurement: This measures a specific metric attribute based on observations obtained from empirical experiments. Various types of empirical evaluation based on the findings from human-in-the-loop experiments have been explored in the existing studies for evaluating information risk for authentication mechanisms (i.e., strength of passwords [Davis et al. 2004; Haga and Zviran 1991]), usability based on users' perceived usefulness (i.e., user acceptance technology on e-shopping on the Web [Shih 2004]), economic cost due to information breach [Campbell 2003], and dynamic threats by network attacks [Peng et al. 2016; Zhan et al. 2013, 2014, 2015].

Metrics in experiments: Experiments often use the following metrics:

— *Atomic metrics* are metrics that quantify the quality for a single dimension of system or application performance [Chen et al. 2018], such as query response time, correct operations for a particular service, correct message delivery, and so forth. The atomic metrics are used as arguments or inputs to evaluate an upper level metric such as trust, security, resilience, and agility analysis methods or models for further upper level reasoning metrics (e.g., uncertainty reasoning).
— *VA effectiveness* measures the number of vulnerabilities identified by the VA process. This metric can have many variants, including the time / resource for VA and the coverage of VA (e.g., which components or procedures of a system are assessed?).
— *PT effectiveness* measures the number of exploitable vulnerabilities identified by PT. The variants of this metric include the characteristics of the vulnerabilities (e.g., the easiness or hardness in exploiting them) and the coverage of PT.
— *Systems overall attack resistance* measures the overall vulnerability and resistance against red team testing.
— *Red team competence* measures the competence of red teams. This models *team competitions*, which challenge teams directly against each other by setting up a system with vulnerabilities and challenging teams to investigate the system. Then each team can be evaluated and compared to each other against a common checklist. The teams should be independent teams; we can compare the effectiveness of teams in identifying the number of vulnerabilities or types of vulnerabilities each team finds.

2.2.3. **Accuracy of Measurements**. We assume that the measurement of a metric can be accurately obtained in the real world. In practice, measurements of metrics are often error-prone due to the uncertainty imposed by unknown attacks or even attackers attempting to disrupt the measurement process. This leads to a question on how trustworthy the measurement process is (e.g., quality of intrusion detection system

with minimum detection errors). The accuracy of measurements can be ensured by: (1) minimizing errors in measurements by learning any changes of system states where high measurement inaccuracy is introduced due to the lack of adaptability to system dynamics; and (2) minimizing the deviation from the ideal reference measurements when a metric cannot be measured directly.

3. PROPERTIES OF A SYSTEM AND ITS QUALITY

3.1. Key Component of Systems

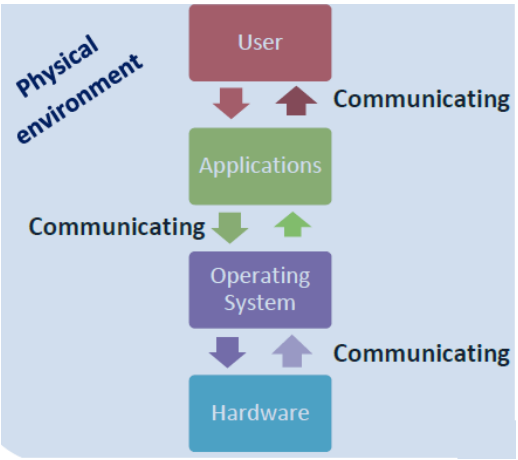

Fig. 1. Computer-based systems consisting of hardware, software, user, network, and physical environment.

In this paper, a 'system' refers to a computer-based system that consists of a set of interacting entities in the context of computing and communication [Avizienis et al. 2004]. A system can be composed of the following multiple factors: hardware, software, network, human factors, and physical environments. In principle, computer-based systems include both enterprise cyber systems and the cyber sub-systems of cyber-physical systems, meaning that STRAM can be equally applied to these broad settings. The interactions of those factors and their effect are critical to the overall system conditions measured by *quality-of-service* (QoS) [Avizienis et al. 2004]. We describe the components of a system as the scope of a computing system in Fig. 1. QoS of a system is affected by threats against the system, including errors, faults attacks, and failures, which are elaborated below.

3.2. Threats

Threat against a system refers to anything that can or may bring harmful effects to the state of the system and lead to improper service states (e.g., erroneous behavior, unavailable service, and/or system shutdown due to a critical failure). In cybersecurity, threats are considered to derive from systemic threats as well as arise from internal agents. The systemic threats are to breach cyberspace safety by creating unintended dangerous situations that introduce unpredictability of computers and information systems [Hundley and Anderson 1995]. Those threats include failures of software and hardware that cannot be fixed in digital technology and/or programming, which is in short called the cause by 'an inherent ontological insecurity within computer systems' [Edwards 1996; Denning 1999].

As highlighted in Fig. 2, Avizienis et al. [2004] classify threats assuming that an active *fault* introduces incorrect service, producing system *error(s)*. If the error is not detected and treated with a proper response, it will cause a system *failure*. This means that any vulnerability leading to a fault or error can be considered as a threat. A vulnerability may come from a system design defect in the process of developing or maintaining the system as well as malicious activities by either inside or outside attackers (e.g., misconduct of users or system operators). That is, a vulnerability may stem from either the unintentional or intentional misconduct. If it is not properly handled, by either detecting or preventing it, the vulnerability leads to a fault and an active fault can trigger an incorrect service. If the incorrect service is not fixed, an error is generated. If the error is critical but not fixed, it causes a system failure [Avizienis et al. 2004]. Thus, a threat includes anything that can or potentially can cause harm to the system, resulting in system failure.

Typically security analysis must be conducted with respect to a clearly defined threat model which includes assumptions in terms of system failure conditions, properties of hardware and software, and the behaviors of users, attackers, and defenders. For static threats (e.g., potential threat due to inherent system vulnerability such as weak cryptographic techniques), analyzing the security of building-blocks, such as cryptographic mechanisms, is sufficient. However, for dynamic threats (e.g., potential threat due to changing environment or dynamic system conditions such as vulnerability introduced by software installation), security analysis is not trivial because if the attack strength changes, the corresponding defense and the level of system security will be affected accordingly. Under the dynamic threat model considering the changing status of the key factors of systems and attack-defense interactions, resilience and agility metrics should capture the time-varying quality of a system.

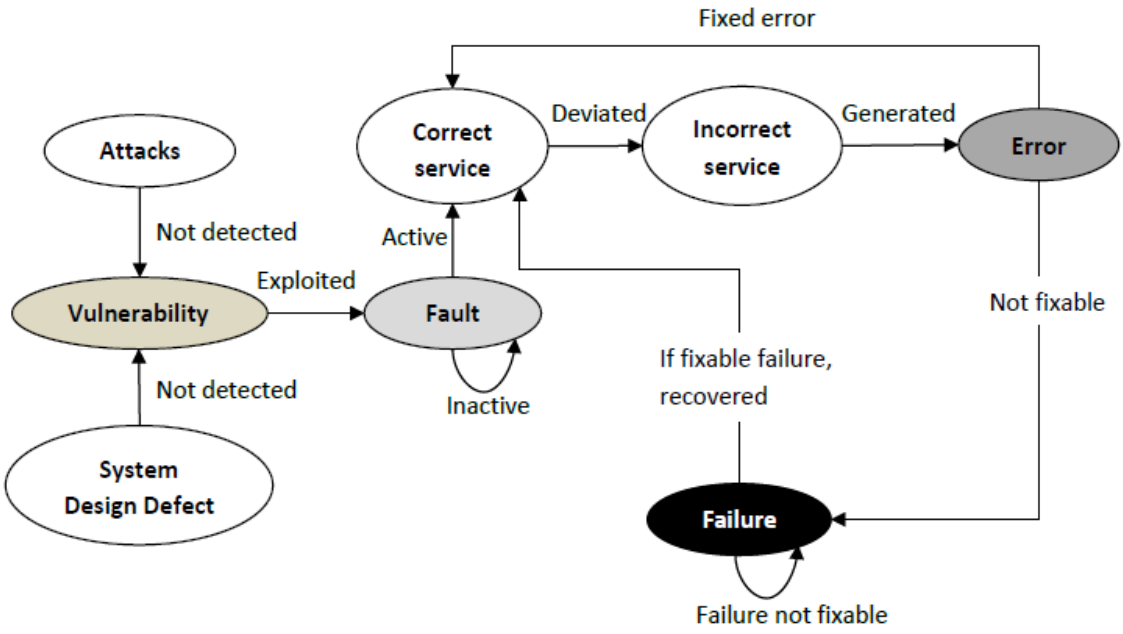

Fig. 2. Relationships between service, threats, vulnerability, and recovery of a system.

3.3. Key Attributes of Trustworthiness

STRAM considers attributes of the trustworthiness of a system in terms of security, trust, resilience, and agility metrics. In this section, we discuss those attributes as submetrics of the STRAM framework proposed by this research. We will discuss various definitions of trustworthiness and their relationship to STRAM. Table I summarizes the definitions of security, trust, resilience, and agility based on the literature.

Table I. Definitions of security, trust, resilience, and agility in various disciplines.

Key submetrics	Definitions	Source
Security	Making all aspects of a computing system, including both physical and cyber systems, free from danger or threats by preserving availability, integrity, and confidentiality	[Pfleeger 2006]
Trust	Subjective belief that a trustee will behave as a trustor expected when taking risk under uncertainty based on the cognitive assessment of past interactions with the trustee	[Cho et al. 2011, 2016]
Resilience	Ability to withstand system degradation by reducing the duration and magnitude of disruptions and by recovering a normal, functional system state persistently within acceptable delay and cost	[Haimes 2009b]
Agility	Ability to deal with unexpected changes or situations (e.g., attacks or errors) while still providing rapid, accurate, proactive, and efficient services	[Alberts 2007; Conboy 2009]

As reviewed in Section 2.1.4, security and dependability have been considered as key system metrics. However, the metrics defined in the literature do not address the multidimensional quality of a system. For example, the existing security or dependability metrics do not consider any subjective judgments provided by human system / security analysts or users while trust metric is capable of capturing Humans' perceived, subjective opinions. Moreover, the existing metrics do not explore attributes such as *agility*, capturing high dynamics in multi-genre domains and many systems dealing with the mixture of information, communication, and/or social-cognitive technologies.

We introduce the concept of 'trustworthiness' in order to indicate a comprehensive, holistic aspect of system quality. The concept of trustworthiness has been used to indicate a good state of quality in describing information, an entity, and/or a system. In social psychology, trustworthiness is also used as a cue to determine whether a person is cooperative or not [Deutsch 1960], representing the 'ability, benevolence, and

integrity' of a trustee [Colquitt et al. 2007]. In philosophy, trustworthiness is defined to indicate the excellence of a person's character [Ivanhoe 1998]. Although in social sciences, the concept of trustworthiness has been used to describe a person's characteristics mainly in terms of integrity, it has been also used to indicate the quality of a high-assured information or system in engineering or information technology domains. Fogg et al. [2001] used the term trustworthiness to describe the key element of information credibility such as 'well-intended, truthful, and unbiased' in terms of the perceived goodness or morality of the source in the context of evaluating the credibility of websites in Internet. More broadly, in networked information systems (NIS), trustworthiness refers to a system state that shows the assurance of performing its required functionalities in the presence of environmental disruptions caused by either natural or human-made errors, hostile attacks caused by either inside or outside attackers, and/or system errors caused by hardware and/or software flaws. Ensuring the trustworthiness of the system is said more than 'assembling components that are themselves trustworthy' [Schneider 1999].

Avizienis et al. [2004] define dependability as a same concept as trustworthiness, assurance, high confidence, or survivability. However, they should be distinguished clearly because the aim of the metric is different. In this paper, we use the term 'quality' and 'trustworthiness' of computer-based systems interchangeably. In addition, although the term 'trustworthiness' is often interchangeably used with trust, trustworthiness is distinguished from trust. That is, trustworthiness refers to an objective aspect of trust based on evidences or observations whereas trust includes subjective aspects of a cognitive entity's opinion, such as that of a human [Cho 2015]. In this work, we use the notion of trustworthiness to represent the overall system quality indicator that should embrace the quality of services, operations, and/or functionalities in the levels of hardware, software, network, human factors, and physical environments. That is, the trustworthiness of a system should reflect the quality of performance and security reflecting their static and dynamic nature. Now we discuss the four attributes[3] of trustworthiness: security, trust, resilience, and agility.

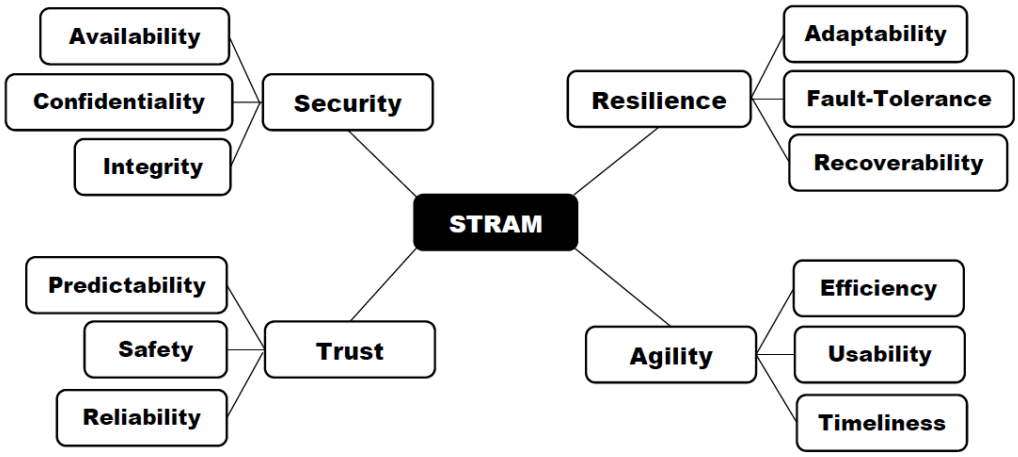

Fig. 3. STRAM ontology with the four submetrics of security, trust, resilience, and agility, under each of which we list three key subattributes.

3.3.1. **Security**. *Security* is defined to ensure "the confidentiality, integrity, and availability of systems, networks, and data through the planning, analysis, development, implementation, maintenance, and enhancement of information systems security programs, policies, procedures, and tools" [Practical Software and Systems Measurement 2006]. In computer science or telecommunication / networking domains, the widely accepted key security goals include confidentiality, availability, integrity, non-repudiation, and authentication. Avizienis et al. [2004] define security as part of dependability, including security and maintainability in which security includes confidentiality, integrity, availability, reliability, and safety.

[3]In this paper, we use a 'metric' to mean a goal to achieve. An 'attribute' is used to indicate a 'submetric' or 'objective' to achieve an upper level metric or goal. For example, while 'security' can be called a metric, 'confidentiality, integrity, and availability' are called 'attributes' to achieve security. Although trust, resilience, and agility can be called metrics (or submetrics), they can also be called 'attributes' to measure trustworthiness.

In this work, we consider integrity, confidentiality, and availability to define security. Unlike the categorization by Avizienis et al. [2004], we include reliability and safety under trust. The security ontology with these three sub-attributes is described in Section 5.2.

3.3.2. **Trust**. The definitions of trust have been discussed by various domains of disciplines in which each discipline defines trust differently [Cho et al. 2011, 2016]. Based on the commonality of trust properties, trust is defined as the willingness to take risk based on subjective belief considering uncertainty and risk as part of the decision making process [Cho et al. 2016]. Trust is often defined in our everyday life by a situation that a trustor trusts a trustee by accepting any vulnerability which is part of the trustee. Similarly, trust between systems is defined as the *accepted dependence* where system A's dependability relies on system B's dependability [Avizienis et al. 2004].

We define trust with three subattributes including *reliability*, *predictability*, and *safety*. The trust ontology with these three attributes is discussed in Section 5.3.

3.3.3. **Resilience**. Resilience is often defined in ecology in order to describe a system state, which is able to deal with fluctuations in an ecosystem [Holling 1973a]. Holling [1973a] defines resilience as the ability to maintain persistence of a system variable state by absorbing the changes to the system variables. Holling also distinguishes resilience from stability in that stability is the capability to return a system to the equilibrium state after temporary fluctuations. Pimm [1984] also clarifies the distinction between resilience and stability in that resilience concerns the speed of recovery while stability does not.

The concept of resilience in the context of systems is defined by "the ability of the system to withstand a major disruption within acceptable degradation parameters and to recover within an acceptable time and composite costs and risks" [Haimes 2009b]. Resilience has also been considered as a synonym for fault-tolerance, which can be a means for achieving system security and dependability [Avizienis et al. 2004].

The concept of resilience in the context of engineering systems [Park et al. 2013; Zolli and Healy 2012] appears to be inherited from the concept of resilience in ecology [Holling 1973a,b]. While it is intuitive to define resilience in engineering systems as the ability of a system adapting to disruptions [Haimes 2009a; Madni and Jackson 2009; Woods and Hollnagel 2006], there have been many variant definitions in different domains. In infrastructure systems, resilience refers to the capability to reduce the duration and magnitude of disruptions. A system's resilience affects its capability in predicting, absorbing, and adapting to the disruptions and its recoverability [Brown et al. 2006].

Resilience has also been defined in social sciences. In economics, resilience means the capability of enterprises and communities to adapt to market shocks and mitigating economic losses in both micro and macro markets [Perrings 2006]. In sociology, resilience indicates the capability of communities to withstand stresses caused by social, political and economic disruptions [Adger 2000]. In organizational behavior, resilience is the capability of organizations to identify risks and deal with perturbations related to their competencies [Woods and Hollnagel 2006]. In addition, national policy makers use resilience to represent preparedness or adaptability of national conditions for rapid recovery from social, economic, or political disruptions [Department of Homeland Security 2015]. The high awareness of national leaders towards the resilience of a system, society, or national situations implies the criticality of dynamic, adaptive responses to any disruptions made to a current system, rather than only the static properties of the system [Haimes 2009a; Madni and Jackson 2009].

Based on the various definitions of resilience mentioned above, we choose three key subattributes of resilience in STRAM, which are *recoverability*, *fault-tolerance*, and *adaptability*. We describe the resilience ontology in Section 5.4.

3.3.4. **Agility**. Agility is defined as "the ability of an entity to be effective in the face of a dynamic situation, unexpected circumstances, or sustaining damage" by emphasizing "the synergistic combination of robustness, resilience, responsiveness, flexibility, innovation, and adaptation" [Alberts 2007]. Agility also means the continual readiness of an entity to respond rapidly, accurately, proactively, and economically continuing to provide high QoS [Conboy 2009]. An agile system is highly proactive, responsive, and quickly recoverable to sudden threats or errors introduced to the system.

In military contexts, agility is treated as "the capability to successfully cope with changes to circumstances" [Alberts 2011]. Alberts discusses the concept of *agility quotient* (AQ) and identifies six enablers of agility [Alberts 2011]: responsiveness (e.g., service response time under time pressure), versatility (e.g., an entity with multiple functionalities), flexibility (e.g., accomplishing a task in multiple ways), resilience (e.g., recovering from degradation or damage), adaptiveness (e.g., adapting to a dynamic environment for survival), and innovativeness (e.g., novel response methods upon sudden changes or under attack). The intuition is that AQ could be tested and measured similar to the Intelligence Quotient (IQ), but the pre-

cise definition of AQ was not given. Since resilience can be an enabler of agility, agility is not orthogonal to resilience.

In the context of cybersecurity, agility is examined as "any reasoned modification to a system or environment in response to a functional, performance, or security need" [McDaniel et al. 2014]. For example, as an intrusion prevention technique, Moving Target Defense (MTD) mechanisms are maneuvers to enhance security in a given environment. The major challenge in using MTD as a maneuver is associated with three factors: cost, secrecy, and side-effect. Recently it has been debated whether to include agility as a key required metric or not [Alberts 2011]. In this work, we decide to include agility as one of key metrics to represent the quality of a system in order to capture the dynamic nature of cybersecurity in trustworthy systems. Although we consider resilience and agility as separate metrics, they are many in common particularly in terms of adaptability to sudden changes or attacks to maintain system availability and reliability. In this work, we consider resilience to more focus on measuring strength against threats while agility captures the speed and cost of adaptability to sudden changes or dynamics. We describe the agility ontology with the three key attributes, *timeliness*, *usability*, and *efficiency* (i.e., complexity or service cost), in Section 5.5.

Table II. Key attributes in the STRAM trustworthiness metric framework, including **S**ecurity [Avizienis et al. 2004; Nicol et al. 2004; Pendleton et al. 2016], **T**rust [Al-Kuwaiti et al. 2008; Avizienis et al. 2004; Cho 2015; Nicol et al. 2004], **R**esilience [Cholda et al. 2009; Haimes 2009b], and **A**gility [Alberts 2007; Conboy 2009; Dekker 2006].

Key metric attributes	Definition	Equivalent or sub-attributes	**S**	**T**	**R**	**A**
Reliability	The state of being reliable by providing same results for what is needed on repeated requests	Predictability; competence; consistency; stability; certainty; fault-forecasting; high-confidence; assurance; survivability		✓	✓	✓
Availability	The state of being present or ready for use	Readiness	✓	✓	✓	✓
Safety	The state of not being harmful or dangerous	Security, protection	✓	✓	✓	
Confidentiality	The state of being secret or private	secrecy, privacy	✓	✓		
Integrity	The state of being honest, fair, sound, complete, or whole	Accuracy; credibility; correctness	✓	✓		
Robustness	The state of being strong and healthy	Fault-tolerance; performability; accountability; authenticity; nonrepudiability	✓	✓	✓	✓
Maintainability	The state of keeping in good condition by fixing problems or repairing	Recoverability; retainability; correctability; self-healing; self-repair			✓	
Adaptability	The state of being able to change to work or fit better	Autonomy; learning; extensibility; reconfigurability			✓	✓
Usability	The state of being used for convenient or practical use;	Automatability; flexibility; learnability; satisfaction; compatibility; reusability; complexity		✓		✓
Timeliness	The state of being at the right time	Quickness; decisiveness				✓
Efficiency	The state of being capable of generating what is needed without wasting resources (e.g., materials, time, or energy)	Leanness; simplicity; scalability				✓
Reactiveness	The state of being readily responsive to a stimulus	Readiness; fault-removal			✓	✓
Proactiveness	The state of being active to prepare for future problems, needs, or changes	Preparedness; fault-prevention				✓

Table II summarizes the key attributes of trustworthiness in terms of security, trust, resilience, and agility based on the literature and the dictionary definitions of each attribute [Al-Kuwaiti et al. 2008; Alberts 2007; Avizienis et al. 2004; Cho 2015; Cholda et al. 2009; Conboy 2009; Dekker 2006; Haimes 2009b; Nicol et al. 2004]. In Table II, *availability* and *robustness* are shown as common attributes of the submetrics of security, trust, resilience, and agility. Security can also be an integral part of trust.

In the proposed STRAM ontology, we consider *safety* as one of the trust attributes where safety consists of *cybersecurity* and *physical security*. Cybersecurity refers to security in STRAM. *Resilience* is often considered the same as recoverability or maintenability (including fault-tolerance). Some dimensions of *agility* are different from those of trust and resilience because agility focuses on measuring how quickly and adaptively a system responds to and functions under sudden changes or attacks, requiring a learning capability under high dynamics. Agility and resilience have many common attributes as described in Table II. Fig. 3 shows the four submetrics of STRAM with each submetric consisting of three key attributes, respectively. We chose the three key attributes under each submetric because they are regarded as most critical aspects to fully cover the measurement of each submetric based on our literature review and insights obtained from it. We believe this can provide a good starting point to define each submetric. Although we will show each submetric's ontology in the following section, we show Fig. 3 in order

to deliver a big picture of each submetric ontology in terms of what attributes are mainly considered for each submetric. The interdependencies between attributes under each key submetric are described in the proposed STRAM ontology, as shown in Fig. 1 of the appendix.

4. EXISTING METRIC ONTOLOGIES

An ontology is "a formal specification of a *shared* conceptualization" [Borst 1997]. Guarino [Guarino 1998] elaborates the term 'conceptualization' as "a language-independent view of the world, a set of conceptual relations defined on a domain space." An ontology can be seen as a language-dependent cognitive artifact committed to *a certain* conceptualization of the world [Guarino 1998]. Therefore, an ontology indicates a set of representational primitives to model a domain of knowledge or discourse. The representational primitives include concepts, attributes of concepts, and relationships between concepts. When ontologies are expressed within a logical framework, we talk about 'formal ontologies'; when formal ontologies are encoded in a machine-readable language, such as the W3C Web Ontology Language (OWL), they become computational ontologies [Guarino 1998].

There have been efforts to develop metric ontologies. Paul et al. [2008] develop an ontology-based assessment framework of trustworthiness including dependability and other attributes. The framework provides automated assessment of trustworthiness for individual system entities and integrates them into an overall integrated system.

The proposed STRAM framework considers the four key attributes needed to measure the trustworthiness of a computer-based system using the ontology methodology. In this section, we give a brief overview of existing ontologies for each attribute considered in STRAM.

4.1. Security Ontologies

Nowadays we have a significant amount of security terminologies but much of the current terminology is too vague and their meanings often overlap with each other. Clear ontologies of security terminologies are solely needed. Donner [2003] defines an ontology as "a set of descriptions of the most important concepts and the relationships among them." An ontology is critical to understanding security related issues and to communicating even between security experts or students / researchers in this field.

Kim et al. [2005] develop a security ontology to annotate resources aiming to discover resources that meet security requirements. The proposed ontology is defined based on the functional aspects of resources such as mechanisms, protocols, objectives, algorithms, and credentials in a different level of granualities. Tsoumas and Gritzalis [2006] use an an ontology to build a security management framework for information systems. The proposed security management ontology provides reusable security knowledge interoperability (i.e., shared meaning of reusable security knowledge), aggregation, and reasoning for security knowledge management.

Parkin et al. [2009] incorporate human behavioral aspects into an information security ontology. The proposed ontology aims to comply with external standards (i.e., ISO27000) while taking into consideration an individuals' security related behaviors within an organization such as password creation. This work is novel in its inclusion of human factors while existing works only focus on system security and organization policy. Blanco et al. [2011] conduct a comprehensive survey on security ontology proposals to identify the key security requirements to be considered to build an integrated and unified security ontology. The key security concepts discovered from the survey include reliability, security protocol, security mechanism, security policy, security risk, security measurement, and security attacks.

The existing security ontologies discussed above are not used for security metrics. Pendleton et al. [2016] introduce the first systematic security metrics with an associated ontology based on the notion of attack-defense interactions [Xu 2014a]. The ontology leads to four classes of security metrics: vulnerability metrics, attack metrics, defense metrics, and situation metrics. Each class of metrics formulates a submetric ontology. Our research goes beyond [Pendleton et al. 2016] by presenting an overarching ontology including the four submetrics of trustworthiness (i.e., security, trust, resilience, and agility). Moreover, the security metrics ontology is based on the classic view of confidentiality, integrity, and availability, which is orthogonal to the view of attack-defense interactions adopted in [Pendleton et al. 2016].

4.2. Trust Ontologies

Ontology-based definitions and models of trust have been studied in various domains [Viljanen 2005]. Chang et al. [2007] propose generic trust ontologies comprising three components for service-oriented network environments of agent trust, service trust, and product trust. Dokoohaki and Matskin [2007] propose a trust ontology to reduce the semantics of a trust network structure for social institutions and ecosystems on Semantic Web. Blasch [2014] discusses many sources to derive trust in a system with six

general areas, including user, hardware, software, network, machines, and the application. He maps trust associated with each area to specific attributes to define the trust ontology. Golbeck and Parsia [2006] present an ontology-based approach to integrate semantic web based trust networks with provenance information to evaluate and filter a set of assertions. Squicciarini et al. [2006] design a reference ontology to develop privacy-preserving trust negotiation systems that allow the secure exchange of protected resources and services by subjects in various security domains. Taherian et al. [2008] enhance the extensibility of the ontology-based trust model encompassing features of pervasive computing contexts.

Different from the existing trust ontologies above [Chang et al. 2007; Dokoohaki and Matskin 2007; Squicciarini et al. 2006; Taherian et al. 2008; Viljanen 2005] focusing on trust in a specific system context, this work proposes a system-level trust metric as the part of representing the trustworthiness of a system.

4.3. Resilience Ontologies

Vlacheas et al. [2011] develop a resilience ontology with several subontologies, including a domain ontology, a threat ontology, a threat agent ontology, a means ontology, and a metrics ontology. The metrics ontology, which adopts the security and dependability metrics presented in [Avizienis et al. 2004], represents the relationship between metrics and resilience in a sense that certain metrics can represent system resilience in part (e.g., availability, reliability, and maintenability). The European Network and Information Security Agency [ENISA 2011] also defines a resilience ontology based on associated domains in terms of metrics, threats, means, threat agents, and domain. However, existing ontologies do not address the attributes of resilience to develop resilience metric.

4.4. Agility Ontologies

Agility is a new and critical attribute of an agile system [Alberts 2014; McDaniel et al. 2014]. Agility has been studied in the enterprise system domain as describing the ability to deal with sudden changes in situations. Kidd [1994] defines agility as a quick and proactive response adapting to unexpected changes in an enterprise system. Yusuf et al. [1999] emphasize the use of reconfigurable resources to achieve speed, flexibility, innovation and quality services. Sherehiy et al. [2007] point out that the key aspects of agility include speed, flexibility, and effective response to uncertainty and sudden changes.

Agility metric ontology research has not been explored in the literature. Very recently, Salo et al. [2016] create an agility ontology and application to measure the degree of agility in an organization based on different characteristics. Our STRAM metric framework includes agility as one of the key dimensions representing trustworthiness and correspondingly an agility ontology as a submetric of STRAM.

5. ONTOLOGY-BASED STRAM AND MEASUREMENTS

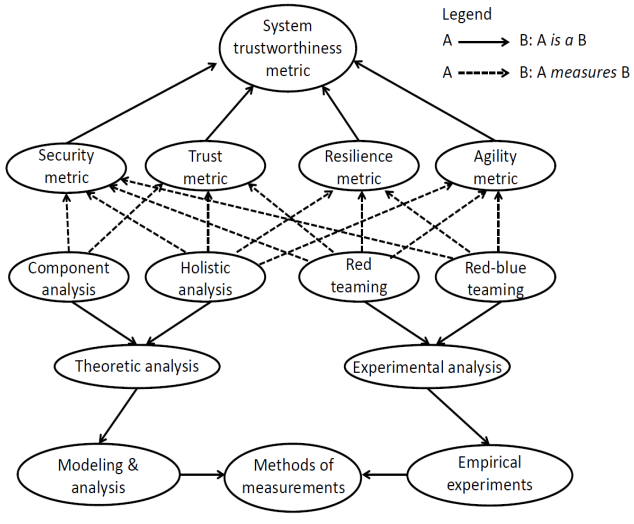

Fig. 4. A metrics-measurement ontology.

In this section, we discuss the overall hierarchical structure of the proposed metric ontology of STRAM, and elaborate on the sub-attributes for each metric within the metric ontology.

5.1. Hierarchical Structure of the Metric Ontology

Fig. 4 describes the high-level ontology of trustworthiness metrics and measurements based on two types of relations: 'is a' and 'measure'. As discussed above, trust, security, resilience, or agility 'is a' trustworthiness metric. Each of the two methods of measurements described in Section 2.2.2 'is a' measurement method. Component analysis (e.g., proving the security of a cryptographic primitive) or holistic analysis (i.e., analyzing the security in terms of system-level perspectives where a system consists of multiple components) belongs to the theoretical analysis as the 'modeling and analysis-based measurement method.'

Red teaming (i.e., defenders do not adjust their defense during the course of red-team attacks) or red-blue teaming (i.e., defenders adjust their defense during the course of red-team attacks) 'is an'

experimental analysis as the 'experiment-based measurement method.'

Component analysis, holistic analysis, red teaming, and red-blue teaming are the concrete methods to 'measure' the trustworthiness of a system in terms of trust, security, resilience, and agility from various angles of system quality. For example, component analysis, holistic analysis, and red teaming are often used to "measure" the trust metric (more specifically, the metrics belonging to the trust submetric), while red-blue teaming would not be applicable because the trust assumptions are rarely changed during the course of red teaming analysis. On the other hand, holistic analysis would be applicable to all of the four kinds of metrics. Recently some studies have taken a holistic analysis approach such as the Cybersecurity Synamics framework [Xu 2014b]. In Sections 5.2–5.5, we describe the ontologies for submetrics considered in STRAM.

5.2. Security Metric Ontology

Aligned with widely used security goals, we propose a security ontology considering the three key security goals, availability, integrity, and confidentiality. We consider non-repudiation and authentication as sub-attributes of confidentiality because they are often associated with access control where some secret information is made available to an authorized user in question.

5.2.1. **Availability**. Availability refers to a system state in which the fully required functionalities are provided without failure or a recovery process. High system availability refers to a long system up-time with high recoverability capability in the presence of failure [ReliaSoft Corporation 2003]. On the other hand, reliability refers to a system state providing the fully required functionalities without any interruptions (e.g., failure) but without recovery process [ReliaSoft Corporation 2003].

— *Data availability* refers to a system state that can provide the data requested by a user regardless of its correctness. The correctness of the data is related to data integrity, which will be discussed below. This can be measured by a probability that a requested data item is available at time t; and
— *Service availability* means a system state that can provide the normal, proper services based on the requested specifics. This can be measured by a probability that a system is available and provides a requested service at time t.

For both data and service availability, the impact of the loss of availability can be estimated based on the consequence of a security breach (i.e., risk) or performance degradation (e.g., unavailable service due to denial-of-service attacks). Well-known availability metrics are Mean Time To Failure (MTTF), Mean Time Between Failure (MTBF), and Mean Time To Repair (MTTR); some of these concepts will be elaborated when we discuss the trust metric ontology below. Measuring system up-time in the presence of attacks is closely related to measuring system resilience. Thus, we will discuss MTTF or MTBF, and MTTF under Section 5.4 when discussing recoverability. Note that when repair is not available (i.e., non-repairable systems), MTTF is used to measure system reliability. On the other hand, when repair is possible (i.e., repairable systems), MTBF means system availability.

5.2.2. **Integrity**. Based on the traditional concept of integrity as one of the security goals, integrity is mainly concerned with data integrity. In this work, we extend the concept of integrity one step further and embrace both service integrity and data integrity.

— *Service integrity* means to what extent a system is providing the correct service. Thus, service integrity is closely related to the degree of software vulnerability or trustworthiness, which can be captured by the following example metrics:
  — *Degree of service vulnerabilities* measures any known or unknown vulnerabilities that can be exploited by attackers aiming to compromise system components or services. These vulnerabilities can significantly affect quality-of-service (i.e., software, hardware, or network) and can disrupt a system's operations. A well-known vulnerability scoring system, called Common Vulnerability Scoring System (CVSS) [The Forum of Incident Response and Security Teams 2015], uses various metrics to measure system vulnerabilities as an attack vector (i.e., vulnerabilities exploitable by attackers in terms of accessibility or complexity), attack complexity (i.e., the effort an attacker has to make in order to exploit a vulnerability), required privilege (i.e., the privilege needed to exploit a vulnerability), and user interaction (e.g., how much legitimate user's cooperation is needed for successful exploitation). In addition, even if software is free from any vulnerabilities, its runtime integrity may be compromised by code reuse or return-oriented programming attacks [Schuster et al. 2015].
  — *Impact of vulnerability exploitation* refers to the consequence in terms of the service integrity when a known or unknown vulnerability is exploited by an attacker. Risk is often used to indicate the

likelihood that a vulnerability is exploited by an attacker. Impact is one of the key factors to assess risk, where risk is often estimated based on a function $f(vulnerability, threats, consequence)$ (i.e., Probability Risk Analysis (PRA) [Jensen 2002], discussed in Section 6.2), although this risk calculation still remains arguable [Brooks 2003]. The example metrics to measure the impact of vulnerability exploitation can be: (1) the number of attack incidents; and (2) the damage caused by the attack.

— *Data integrity* ensures the integrity of given data, measured by the probability that the data are kept intact at time t without being altered, corrupted, or destroyed by any system or user errors, including hardware or software errors, malicious parties, or unintentional user mistakes. Data integrity can consist of preserving the following attributes:
  — *Correctness* is to ensure whether the data are kept intact without being modified, forged, corrupted, or destroyed by a third party. Data correctness is often ensured by cryptographic keys together with some message integrity mechanisms (e.g., message authentication code or digitial signatures), data replication (or mirroring), or RAID (Redundant Array of Independent or Inexpensive Disks) parity [Sivathanu et al. 2005]. The effectiveness of these mechanisms can be used as a metric to measure data correctness in terms of a correctness detection ratio or the impact of loss of data correctness. Note that even if some of data is missing (i.e., incomplete data), the remaining data can be checked for their correctness.
  — *Completeness* means that all the data required should be available in a stable state [Pipino et al. 2002]. This can be measured by the amount of missing information and its impact. Note that even if we obtained complete data with all required data (e.g., required information in a certain form such as name, phone number, address), but the data may not be correct. Hence, correctness and completeness of data should be evaluated separately.
  — *Validity* is to ensure data correctness in terms of logical reasonableness which checks data type, range or constraint, code and cross-reference, or data structure [Pipino et al. 2002]. This can be measured by the amount of invalid information and its impact.

— *System integrity* is used as a term to indicate maintaining system operations without interruptions by compromised system components including hardware, software, and user as follows:
  — *Hardware integrity* can be estimated by various types of software to check memory, CPU, or file systems [Wang et al. 2010]. Thus, the integrity of the hardware-checking software is also critical to receiving correct information for the status of hardware integrity;
  — *Software integrity* can be verified by various types of software integrity verification tools that can ensure software's code integrity, correctness, coverage, complexity, or exception handling [Adrion et al. 1982]; and
  — *User integrity* can be measured by the degree of malicious activities a user shows while using a system. User integrity also belongs to the user reliability metric which is discussed in 'reliability' of the trust ontology metric in Section 5.3;

5.2.3. **Confidentiality**. Confidentiality means that certain information should be available only to authorized parties. Confidentiality can be measured by the following.

Preservation of confidentiality. This measures how often, and to what extent, confidentiality is violated in terms of time and the degree (e.g., criticality). The example metrics may include the following:

— *Secrecy violation* can be captured by (1) the time elapsed after a private key is compromised in Public Key Infrastructure [Guan et al. 2015; Harrison and Xu 2007; Xu and Yung 2009], or (2) the time elapsed without changing a symmetric key upon any membership changes, violating backward or forward secrecy [Xu 2007].
— *Privacy violation* refers to a situation where private information is leaked out to unauthorized parties or without an owner's consent. Social engineering attacks are often culprits of intruding privacy (e.g., phishing attacks). This can be captured by the number of messages leaked out to unauthorized parties or social adversarial users exploiting private information for their own purposes [Cho et al. 2016].
— *Access right violation* measures how often and to what extent the user violates a given access right when a user is authorized to access system resources.

Authentication. The purpose of authentication is to secure a system by ensuring accessibility based on a certain security process. This can be measured by (1) the strength of a user's authentication mechanisms (e.g., weak passwords, stolen passwords or compromised keys); and (2) the impact of the compromised passwords or cryptographic keys.

Non-repudiation. Non-repudiation is to ensure the authenticity of an identity or a signature by proving the integrity of the source of information. It can be measured by the impact of breaking non-reputation assurance by exploiting compromised defense mechanisms (e.g., compromised private key) [Dai et al. 2012; Xu and Yung 2009]. This can be estimated based on a probability that non-repudiation is not properly serviced at time t due to identity attacks (e.g., fake identity attacks).

An interesting interplay exists between data integrity, availability, and confidentiality. Data integrity can adversely affect data availability because deletion of given data makes them unavailable at time t. In addition, when a key is compromised, the confidentiality of information cannot be assured if a third-party uses such keys to access authorized information.

We summarize the attributes of the security metric ontology in Fig. 5. In Fig. 5, $a \rightarrow b$ represents that b is an attribute of a, implying that a is measured by the attribute of b. For example, confidentiality, availability, and integrity are the attributes of security and can measure security while data availability and service availability are also sub-subattributes of a subattribute of security, which is availability, and can measure availability. A 'thick' arrow represents the higher layer of the relationships between a submetric and its attributes in the hierarchical ontology. This notation will be used also for the rest of ontologies for each submetric shown in Figs. 6–9.

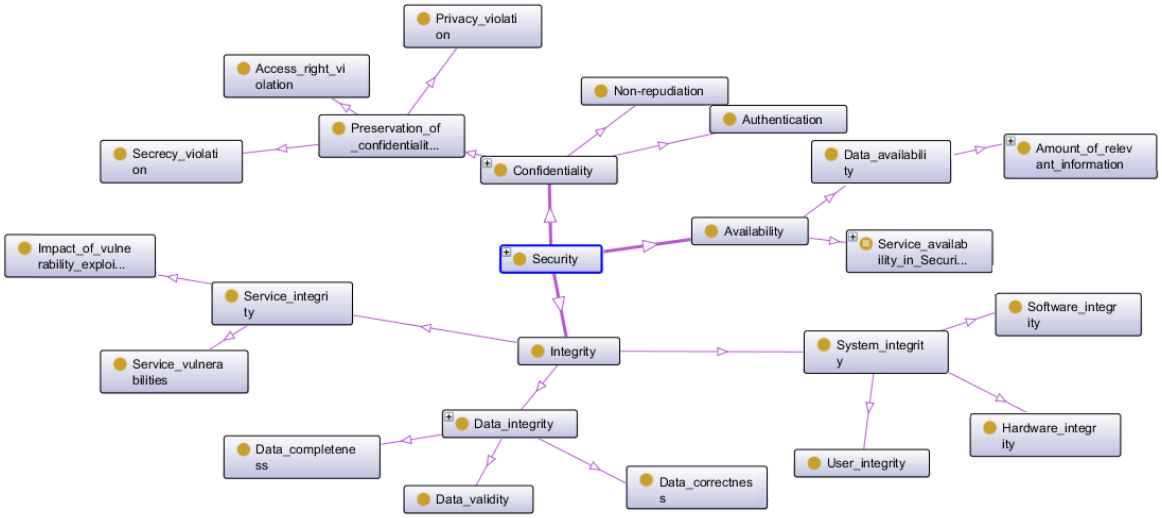

Fig. 5. Security metric ontology.

5.3. Trust Metric Ontology

As addressed in Section 3.3, trust has been defined differently by various domains with different emphasis of concepts [Cho et al. 2011, 2016]. As our target entity to measure trust in a computing system, we choose three main attributes as submetrics to measure trust: reliability, predictability, and safety.

5.3.1. **Reliability**. As mentioned in Section 5.2, Security Metric Ontology, reliability is a system state that can provide fully required functionalities without any failure. Thus, if maintainability (e.g., recoverability as a resilience metric) is high, reliability can increase but availability stays constant [ReliaSoft Corporation 2003]. We categorize system reliability in terms of the following three types of reliability: data reliability, service reliability, and user reliability.

Data reliability. Data reliability embraces all attributes of data integrity discussed in Section 5.2 which include correctness, completeness, and validity. As additional attributes, we consider consistency, freshness, and source credibility to measure data reliability.

— *Consistency* can be checked in the process of monitoring whether data are consistent in structure, space and time. In a database, data should be stored in a consistent structure and with backups maintained at regular time intervals [Haerder and Reuter 1983]. This can be captured by (1) delay occurred to synchronize the views in multiple locations from a previous update to the latest update; and (2) data format entered to maintain the consistent format.

— *Freshness* refers to how recently data are updated from the current time [Wang et al. 2013]. This can be estimated based on the elapsed time from the recorded (i.e., updated) time to the current time.
— *Source credibility* ensures the trustworthiness of an information source in order to evaluate credibility of the information itself. Provenance information is often used to check if information is trustworthy or not based on the assumed interdependency between quality of information and quality of the information source [Cho and Chen 2016; Xu et al. 2010]. Many peer-to-peer trust estimation models have been used to ensure the source credibility based on provenance data [Cho and Chen 2016; Cho et al. 2016; Wang et al. 2013; Xu et al. 2010].

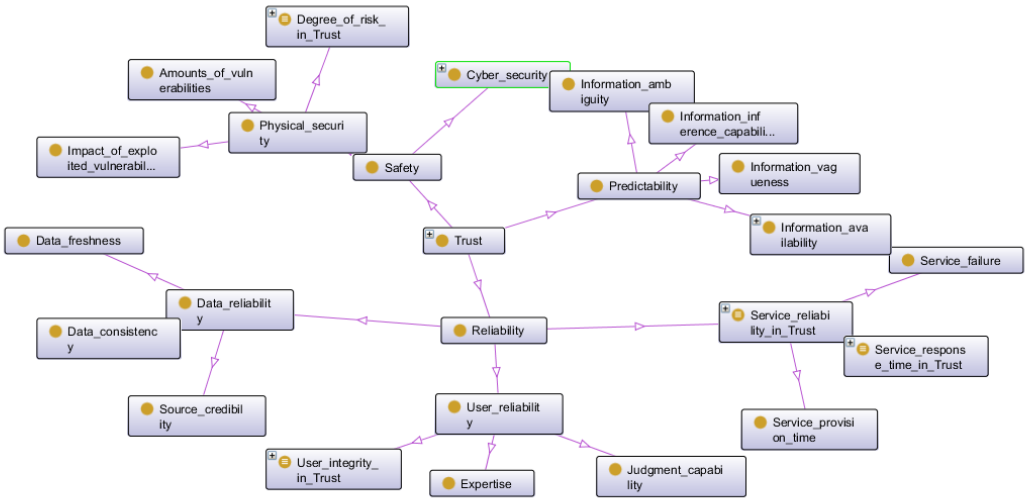

Fig. 6. Trust metric ontology.

Service reliability. Service reliability refers to whether a system is providing proper services without interruptions. We consider service reliability in terms of the services or proper operations provided by hardware, software, and networks. Typical metrics that measure service reliability are:
— *Reliable service provision time* is the time period of providing proper services without interruptions. MTTF is often used as a reliability metric in terms of the period a system properly functions until time t without experiencing any failures. MTTF is also mentioned as a robustness metric under the resilience metric ontology in Section 5.4.
— *Frequency of service failure* counts the number of cases a requested service is not provided to the desired quality.
— *Service response time* measures how quickly a requested service is provided. This is affected by hardware capacity, algorithmic complexity in software, or network delay.

User reliability. User reliability measures how reliably a user, an analyst, or a system operator uses a system, and can be measured by:
— *Expertise* is related to how much experience or domain knowledge a human user has.
— *Judgment capability* is related to a human user's cognitive ability to understand a situation, learning and being adaptive to the changes, or detecting errors.
— *Integrity* is related to a human user's intention; whether the intention is malign or benign. The user may fail a system by mistakes or can intentionally fail the system with a malicious intention (e.g., inside or outside attackers).

5.3.2. **Predictability**. High predictability towards the trustee's future behavior implies high trust in successful performance in a given task by the trustee. Thus, high predictability means high certainty towards the trustee's decision or behavior. The degree of *certainty* is affected by a lack of information or knowledge, vagueness of information, or ambiguity of information. The degree of the causes can be measured by the following metrics:

— *Information availability* can be computed based on the amount of relevant evidence available.
— *Information inference capability* refers to the ability of a used inference tool in terms of how well errors are detected and accordingly correct decisions are made with the used tool.
— *Information vagueness* measures how distinctively the evidence is classified into a certain category to make a decision. The distance (or similarity) between evidence can be used as a metric to measure the degree of information vagueness.
— *Information ambiguity* captures the amount of conflicting evidence which hinders a timely decision.

These metrics are related to each other. The information availability metric measures the amount of evidence available, regardless of the quality. Information vagueness and information ambiguity are different in that the presence of vagueness does not necessarily mean the presence of ambiguity, and vice versa.

5.3.3. **Safety**. Some studies claim that system safety aims to make a system free from hazards which can be caused by unintentional design flaws or other malfunctions [Burns et al. 1992]. System design flaws or malfunctions can also be viewed as system vulnerabilities that can potentially be exploited by attackers. Based on the recent study [Bahr 2014], preserving safety is a must for risk analysis. As discussed previously in Section 3.2, threats may stem from either unintentional mistakes or intentional malicious activities. In that sense, we define safety as the loss of risk or threats, which can be defined by the following three attributes:

Cybersecurity (CS). Cybersecurity is measured by security metrics as described in Section 5.2. This overlapping attribute is described in the STRAM ontology of Fig. 1 in the appendix. CS is closely related to physical security as physical destruction of a system brings loss of information. However, these days, information can also be easily stolen by various types of cyber attacks without accessing physical assets [Weingart 2000].

Environmental security (ES). Environmental security is the protection of the location where a system resides by having security guards, badge readers, cameras, or access control policies [Weingart 2000]. However, as more layers of security policies or procedures are added to enhance environmental security, there may be performance degradation (e.g., long delay to obtain an approval).

Physical security (PS). Physical security is concerned with protecting the physical computing system by placing a hurdle to deter unauthorized physical access [Weingart 2000]. PS is complementary to ES along with logical security (LS). LS is to safeguard software in a computing system by providing user identification, authentication, or access control. In STRAM, LS refers to cybersecurity which is security metric, described in Section 5.2.

The degree of cyber, environmental, and physical security can be measured by the following metrics:
— *Amount of vulnerabilities* measures the frequency of vulnerabilities identified in a system.
— *Degree of risk* measures the likelihood that known or unknown vulnerabilities result in errors by intentional malicious activities (i.e., internal or external attacks) or unintentional mistakes.
— *Impact of vulnerabilities* measures the degree of negative consequences caused by exploited or exposed vulnerabilities (e.g., stolen critical, confidential information). The impact of a vulnerability is naturally linked to how critical the vulnerability is.

We summarize the attributes of the trust metric ontology in Fig. 6.

5.4. Resilience Metric Ontology

Based on the comprehensive survey on the definitions of resilience as discussed in Section 3.3.3, we define three classes of resilience metrics: fault-tolerance, recoverability, and adaptability.

5.4.1. **Fault-Tolerance**. Fault-tolerance refers to the degree of a system's functional state by providing proper services in the presence of threats including faults, errors, or attacks. Fault-tolerance can be measured by two attributes as described below.

Robustness. Robustness measures the degree of attacks that can be tolerated by a system in terms of security or service degradation. The robustness can be captured by the following two attributes.
— *System survivability* refers to the degree of how long a system can normally operate by providing normal system operations. In a system consisting of multiple components (e.g., multiple computer devices), this can also represent how much the system can tolerate attacks or faults even without a complete set of system components available. Multiple metrics have been measured based on the following measurement:
  — *Mean Time To Failure* (MTTF) [Cho 2015] measures the time until a system reaches a failure state based on a number of functional entities in the system. In particular, MTTF measures system sur-

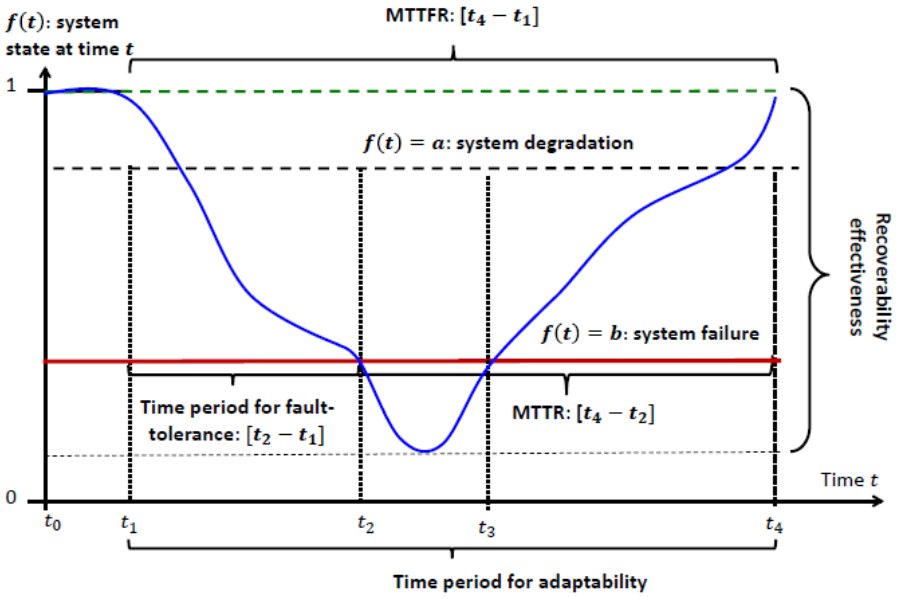

Fig. 7. Measurements of resilience metric.

vivability, if a system is non repairable system, availability is the same as system reliability [Rausand and Hoyland 2009].

- *Percolation threshhold* [Barabási 2016] represents the threshold of the fraction of nodes that can be deleted without network failure because of redundant edges existing in the network. This metric represents network robustness based on the size of a large connected component [Schneider et al. 2011].
- *Network interdependency* [Havlin et al. 2010] is a measure of how interdependent system components affect system failure, including the cascading effect caused by the interdependency.

— *Service reliability* overlaps with the reliability subattribute of the trust metric ontology described in Section 5.3. This interdependency is shown in Fig. 1 of the appendix.

Diversity. This captures the degree of diversity in system components and security mechanisms as follows.

— *Diversity of system components* refers to the diversity of hardware and software. The diversity of the system components is vital because the catastrophic damage of many cyber attacks is caused by the monolithic hardware and software stack. For example, when many devices run the same operating system or application, a single vulnerability in the operating system or application can result in them sharing vulnerabilities. Diversity of software or hardware can be measured by:
  - *The number of versions* of hardware or software components of the same function;
  - *Similarity between versions* measuring the degree of similarity between two versions of the same hardware or software; and
  - *Diversity-enhanced security* estimating security gained by using heterogeneous system components.

— *Diversity of security mechanisms* refers to how diverse security mechanisms can be used to defend against attackers. Diversity can be alternatively attained by Moving Target Defense (MTD) techniques to change attack surfaces [Hong and Kim 2016; Han et al. 2014]. Moreover, deception techniques are also used to confuse or mislead attackers [Stech et al. 2016]. The degree of using diverse security mechanisms can be an indicator to measure part of system robustness.

— *Redundancy of same functionalities* increases reliability by having multiple implementations of a same module [Dove 2005].

5.4.2. **Adaptability**. Adaptability refers to the ability to change itself or resources in response to the changed situations or environments in order to maintain a system's normal operations by providing normal services [Folke 2010; Walker et al. 2004].

Autonomy. Antsaklis et al. [1989] define autonomy in the context of autonomous control systems which function well under high uncertainty for a long period of mission duration and tolerate any failures (or

attacks) without any intervention from outside. Zeigler [1990] defines the concept of autonomy based on multidisciplinary perspectives derived from artificial intelligence, intelligent control, simulation, and robotics with the following three capabilities: (1) capability to achieve given objectives; (2) adaptability to key environmental changes; and (3) ability to develop its own goals. Huang et al. [2004] also address the concept of autonomy in terms of a more broad sense considering the system's ability to 'perceive, plan, decide, and act' to attain given goals.

In this work, we are interested in how autonomously an individual entity can take actions or make decisions based on its own cognitively perceived situation awareness in terms of risk and utility in decision making. System automation can also be an indicator of judging the level of system autonomy [Woods 1996]. We focus the following three aspects of the system to measure autonomy.

— *Degree of local decisions* can represent the degree of an entity's decision making based on its own observations and rules to follow without relying on a centralized third party. Self-determination or self-organization based on local observations can reflect the degree of autonomy [Dove 2005];
— *Degree of intelligent decisions* refers to the degree of rational decision makings to preserve system goals such as maximizing effectiveness or efficiency; and
— *Degree of automation* refers to high efficiency of decision making without human intervention.

Learnability. Learnability represents learning towards dynamic situations in order to make adaptive decisions for improved performance compared to the past performance. In this work, we consider learnability in terms of the following three aspects:

— *Learning defense strength* can be measured based on cost-effectiveness of defense mechanisms and can be learned based on the following:
  — *System vulnerable period* can represent how a system is being protected by current defense mechanisms;
  — *Defense effectiveness* indicates the performance of defense mechanisms used in the system (e.g., detection errors for intrusion detection systems, namely IDS; robustness of encryption mechanisms; or secrecy violation or amount of data leakage using key management techniques);
  — *Number of uncompromised components* indicates the level of system defense in terms of how many components (e.g., nodes) operate properly by providing normal, functional services; and
  — *Defense cost* includes any cost associated with employing defense mechanisms in the system, including defense mechanisms' efficiency as well as financial cost in deploying defense mechanisms to maintain a certain level of system security.
— *Learning attack patterns* can be obtained in terms of the following attack behaviors:
  — *Attack tactics* refer to an attacker's methods for evading defense mechanisms;
  — *Attack intent* indicates an attack's ultimate goal based on connecting attack signatures with attack narratives [Mireles et al. 2016];
  — *Attack types* are a set of attacks available to attackers based on exploitable vulnerabilities;
  — *Attack cost* includes time or resource that attackers need to invest to penetrate into and disrupt a system;
  — *Attack resources* refer to an amount of resources an attacker can utilize to perform attacks (e.g., a size of botnets used by an attacker); and
  — *Attack power* is the capability that an attacker can successfully launch an attack and attain its goal by exploiting targeted vulnerabilities.
— *Learning system vulnerabilities and risk* beyond known vulnerabilities and exposed system risk can be estimated based on identified unknown vulnerabilities exploited by attackers and its impact.

Reconfigurability. We capture the degree of system reconfigurability based on the delay, accuracy, and flexibility associated with the system reconfiguration as follows:

— *Reconfiguration delay* measures how long a system takes to reach a reconfigured state which provides normal operations or services. This is a key metric to measure the degree of reconfigurability.
— *Accuracy of reconfiguration* estimates how reliably the system operates under the changed configurations.
— *Flexibility of reconfiguration* measures whether multiple alternatives exist to reconfigure the system under the changes or attacks. An example can be attack mitigation by adjusting the security policy, security architecture, or security/defense mechanisms to prevent or detect attacks. Elastic capacity, such as decreasing or increasing the level of resources, is also used to reflect flexibility to enhance adaptability [Dove 2005].

5.4.3. **Recoverability**. We consider two classes of recoverability metrics including *recovery delay* and *recovery effectiveness*. In Fig. 7, we use two types of system thresholds to determine a system degradation

state (i.e., system degradation threshold a) and a system failure state (i.e., system failure threshold b) caused by threats, including faults, errors, or attacks, respectively.

Recovery delay. This refers to how long a system takes to reach a normal operation system state from a disrupted system state. We discuss three types of recovery delay as follows:

— *Mean Time To Full Recovery* (MTTFR) captures the whole delay from the time a system starts degrading from a perfect normal operation state to the time the system recovers back to the normal operation state. In Fig. 7, the overall recovery time corresponds to the interval $[t_1, t_4]$.
— *Mean Time Between Failures* (MTBF) is measured by the system uptime between the system failure states. Although Fig. 7 shows only one time failure, MTBF can be estimated based on the sum of intervals $[t_0, t_2]$ and $[t_3, t_4]$. MTBF is calculated by:

$$MTBF = \frac{\sum_{f \in F} \left(u_{s,f} - d_{s,f} \right)}{|F|} \tag{2}$$

where d_s is the start of the downtime and u_s is the start of the uptime, and F is the set of failures for any failure f.
— *Mean Time To Repair* (MTTR) is a fundamental measure of maintainability of repairable items. It is estimated by the average delay between the system failure state and the system's full recovery state which can be computed from the interval $[t_2, t_4]$ in Fig. 7.

Recovery effectiveness. This refers to how much a system has recovered back or close to the normal operation state. Similar to the recovery delay, we discuss the recovery effectiveness based on two system thresholds as follows:

— *Degree of recovery from system degradation* indicates the degree of system recovery from the system degradation state to the current recovered state and can be estimated by $f(t) - a$, where $f(t)$ refers to the current system state, as shown in Fig. 7.
— *Degree of recovery from system failure* refers to the degree of system recovery from the system failure state to the current recovered state, computed by $f(t) - b$ as shown in Fig. 7.
— *Resilience curve* is a measure of the best achievable performance of a system in the worst-case loss scenario as a function of the disruption "magnitude" (e.g., the number of components that are lost at the same time) [Alderson et al. 2013].

Recovery cost. This includes any costs related to recovery process to reach a normal operation system state including the following costs:

— *Intrusion response cost* is the cost associated with actions taken to respond to detected intrusions such as rekeying cost, patching vulnerabilities, or cleaning infected entities;
— *Replacement cost* is the cost to replace hardware or software including financial cost, time, or complexity;
— *Physical storage repair* is the cost to fix/replace destroyed physical infrastructure; and
— *User training cost* is the cost of users' security training or system reconfiguration training upon the occurrence of infected systems.

We summarize the attributes of the resilience metric ontology in Fig. 8.

5.5. Agility Metric Ontology

In STRAM, agility is used as a metric to capture how fast, efficiently, and effectively a system can adapt to unexpected changes or attacks. In this sense, we consider three attributes, 'Timeliness,' 'Usability,' and 'Efficiency' and discuss how they can be measured in the levels of decision making associated with agility process, policy, architecture, and mechanisms (or protocols) for a given system.

5.5.1. **Timeliness**. In the systems engineering domain, agility refers to the *response ability* of a system, including both reactive and proactive responses [Dove 2005]. In this sense, we measure timeliness of services or defense actions to represent agility. Two subattributes are considered as:

— *Service availability* is the same attribute under the availability described in the security metric ontology (see Section 5.2); and
— *Quickness* refers to service response time including both proactive and reactive responses and can be measured by:
  — *Detection time to trigger agility mode* is the time that a system detected an unexpected event.
  — *Decision time to trigger agility action(s)* is a decision delay between the detection time and the decision time for the agility process to be triggered or not.

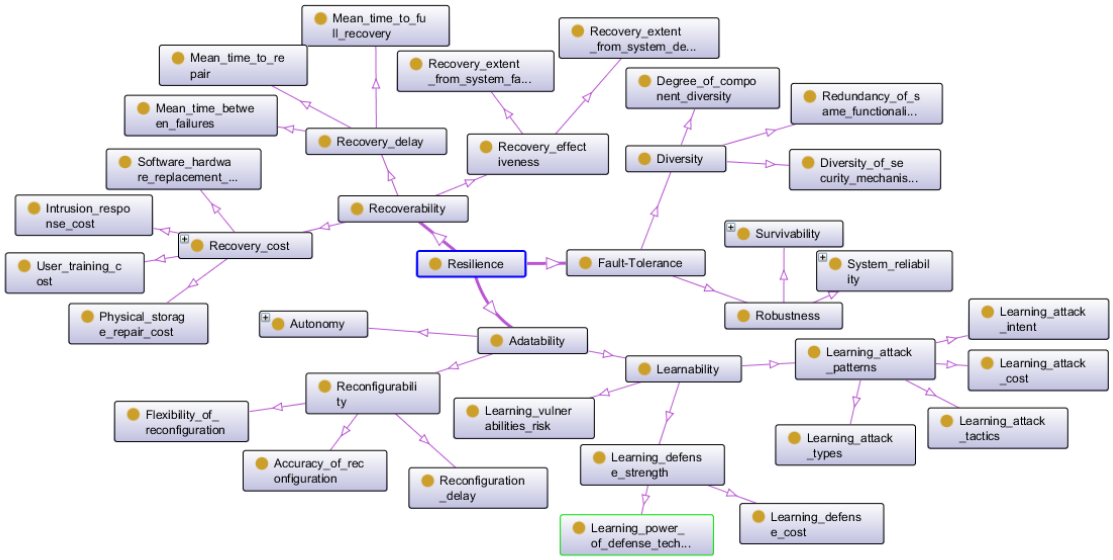

Fig. 8. Resilience metric ontology.

— *Overall agility quickness* is the duration from the detection time to completing the agility process. This metric overlaps a recoverability metric, mean time to full recovery or MTTFR, discussed in Section 5.4.

For agility actions, a system may choose various types of system policy, architecture, or mechanisms as an alternative solution to quickly deliver normal services under the normal system operations.

5.5.2. **Usability**. A well known definition of usability is by Nielsen [1993]: 'usability is about learnability, efficiency, memorability, errors, and satisfaction.' But the main reference of the definition of usability is based on one by ISO 9241-11 [ISO/IEC 1998] which addresses the guidance on usability in terms of "the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency, and satisfaction in a specified context of use." In this work, we define usability as:

— *Easiness* means how easily a user can learn to use a tool such as easiness to fix errors, use defense tools, or to memorize how to use the tools.
— *Service satisfaction level* means how much a user satisfies the provided service in terms of the following three aspects:
  — *Service reliability* is to ensure correctness and consistency of provided services in the presence of repairing or defense mechanisms that are simultaneously being conducted. This can be measured by the effectiveness of different levels of alternatives used (e.g., policy, architecture, or mechanism) to deal with improper system functionalities or attacks. Example metrics can be: (1) *vulnerabilities exposed by using a new alternative* estimated by (i) whether or not an attacker knows vulnerability of the new alternative solution in terms of policy, architecture, or mechanism used; and (ii) whether or not there exists any side-effect by introducing the new solution (e.g., system performance degradation or security holes); and (2) *a lack of service quality* which indicates whether or not expected normal services are being provided in terms of service availability, timeliness, quality-of-service in correctness and consistency, or functionalities.
  — *Usefulness* needs to measure the necessity and convenience of the service, possibly via (1) *frequency of use* indicating how often a user uses a certain application or service; and (2) *easy accessibility* as an indicator of how conveniently a given system provides service for users (e.g., direct access to a particular application or service, simple authentication process such as fingerprint authentication in mobile devices).
  — *Flexibility* means how flexibly a system is in choosing alternative solutions to handle unexpected changes or events. This can be measured by how diverse and sufficient alternative resources (e.g., policies, architectures, or mechanisms) are available in a system. This metric tells us what solutions are available and how they can be used to reconfigure and repair a system as a result of unexpected

changes or events (e.g., attacks). This is also linked to the flexibility under the resilience metric ontology in Section 5.4.

5.5.3. **Efficiency**. Efficiency refers to how efficiently a given service is provided in terms of service cost and complexity to achieve an agile system.

— *Service cost* includes any cost associated with building an agile system including:
  — *Defense cost* includes cost to deploy various alternative solutions (e.g., policies, architectures, or mechanisms) in terms of technical complexity (e.g., efficient module compatibility based on standards [Dove 2005]), performance degradation (e.g., a lack of service availability during the deployment time) and budget (e.g., security investment and cost-effective security investment). This metric is discussed as *response cost* [Dove 2001] or *maneuver cost* [McDaniel et al. 2014] in the literature.
  — *Decision cost* refers to administrative delay, complexity, and effort incurred by the agility decision-making process; and
  — *Recovery cost* includes the cost described under the recoverability attribute of the resilience metric ontology in Section 5.4.
— *Service complexity* refers to (1) easiness discussed in usability; and (2) computational complexity (e.g., algorithmic complexity).

We summarize the attributes of the agility metric ontology in Fig. 9.

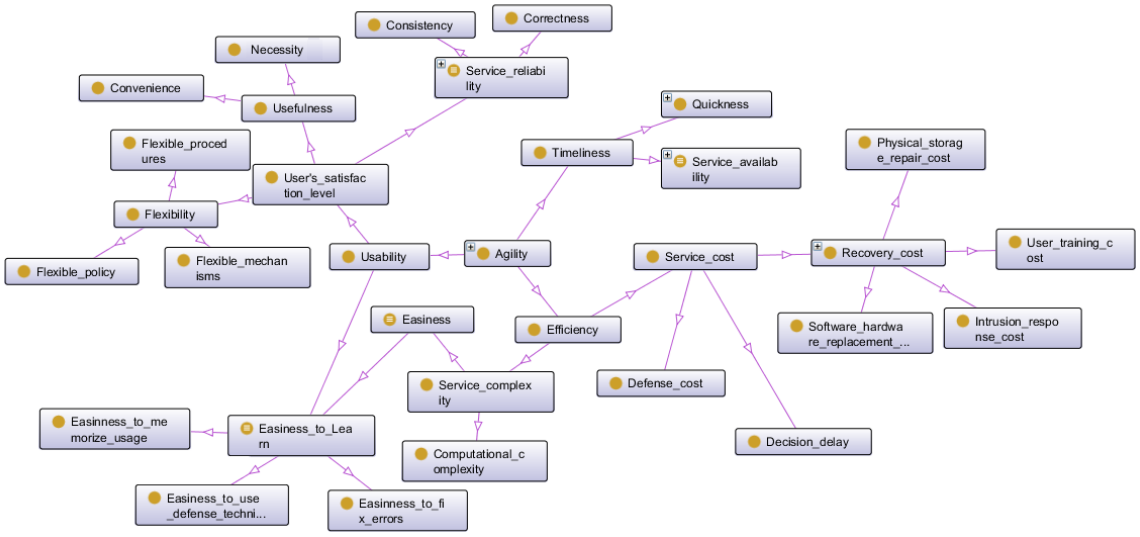

Fig. 9. Agility metric ontology.

Table III summarizes how each submetric can be measured in practice based on the example metrics discussed in Sections 5.2–5.5.

5.6. Aggregating Metrics According to the Hierarchical Structure

As mentioned in Section 5.1, there is a hierarchical structure between the security, trust, resilience, and agility metrics discussed above. As illustrated in Fig. 11 in Section 8, submetrics can be aggregated into a higher-level metric (e.g., the security submetrics may be aggregated into a security metric), and eventually the security, trust, resilience, and agility metrics can be aggregated into a single metric of trustworthiness. Although the aggregation illustrated in Fig. 11 is intuitive, we caution that proper aggregation functions are elusive and more research investigations need to be conducted (e.g., average or weighted-average can be misleading [Savage 2009]).

6. METRICS OF ASSESSMENT TOOLS

In this section, we discuss how to measure the quality of a system by various assessment tools. In this work, we discuss Vulnerability Assessment (VA), Risk Assessment (RA), and Red Teaming (RT).

Table III. Example measurements of the four key submetrics in STRAM.

Submetric		Key measurement	Example metrics
Security	Availability	Service & data availability	MTTF, MTBF, MTTR,
	Confidentiality	Preservation of confidentiality, violation of non-repudiation & authentication	Secrecy violation, privacy violation, access right violation, impact of breaking authentication and key compromises,
	Integrity	Integrity of services, data, systems	CVSS based vulnerability assessment, risk assessment, attack success probability; integrity check of CPU, memory, and file systems; software integrity check (e.g., vulnerable code checking);
Trust	Predictability	Uncertainty quantification	Information availability, accuracy of information inference, information vagueness, information ambiguity
	Safety	Security in cyberspace, environments, and physical systems	Enhanced security based on multiple layers of physical protections, security policies, risk & vulnerability assessment
	Reliability	Reliability of data, services, & users	Data consistency, freshness, source credibility, reliable service provision time (e.g., MTTF), frequency of service failures, service response time, correct decision making rate
Resilience	Adaptability	Autonomy, learnability	Degrees of local decisions, intelligent decisions, or automation; learning defense strength or attack patterns
	Fault-tolerance	Robustness, diversity	System survivability (e.g., MTTF, percolation threshold, network interdependency), service reliability, diversity metrics (e.g., a number of software/hardware versions or multiple modules implementing a same functionality)
	Recoverability	Recovery delay, effectiveness, or cost	MTTFR, MTBF, MTTR; degree of recovery from system degradation or failure, resilience curve, recovery cost (e.g., intrusion response cost, replacement cost, physical storage repair cost, user training cost)
Agility	Timeliness	Service availability or quickness	Detection / decision time
	Usability	Easiness, service satisfaction level,	Service reliability, usefulness (e.g., frequency of use or easy accessibility), flexibility
	Efficiency	Service cost or complexity	Defense cost, decision cost, recovery cost, easiness, or computational complexity (i.e., algorithmic complexity)

One example application of these metrics would be to incorporate them into the U.S. DoD's cybersecurity test and evaluation guidebook [U.S. DoD 2015], which defines the following cybersecurity test and evaluation phases but without giving metrics: (1) understanding cybersecurity requirements; (2) characterizing the attack surface; (3) identifying system vulnerabilities; (4) adversarial developmental test and evaluation; (5) operational vulnerability and penetration assessment; and (6) adversarial assessment of mission execution.

6.1. Metrics of Vulnerability Assessment Tools

Two types of security faults can be defined as 'design flaws' and 'implementation bugs.' *Design flaws* are associated with the architecture of software systems (e.g., lack of access control), while *implementation bugs* are software code-level problems (e.g., cross-site scripting, SQL injection) [McGraw 2006]. Many VA tools use both passive scanning and active scanning.

The purpose of VA is to detect these security faults through the entire lifecycle of a system. If the faults are detected at earlier phases, better defense strategies can be employed. At the *software testing* phase, various VA tools can be used, including penetration testing tools, software inspection and reviewing, static code analysis, dynamic code analysis, or runtime program anomaly detection [Stuttard and Pinto 2007]. At any point in time before or after the system is deployed, penetration testing and red teaming can be used to assess vulnerabilities. Typical procedures are: (1) determining the software programs and network services (e.g., versions or patch levels) running on a target computer; (2) identifying their vulnerabilities; and (3) verifying whether the vulnerabilities can be exploited or not [Goertzel and Winograd 2011].

In general, VA methodlogies are categorized by two main classes: 'manual' and 'automated' [Chess and West 2007]. Both methodologies can be used at any phase of the system lifecycle with the following four steps:

— An architectural analysis to identify the structural components of a system, such as hosts, processes, and threads;
— A resource analysis to identify the resources accessed by the components as well as their operations on the resources;
— A trust and privilege analysis to identify the trust assumptions and privilege level on which an executable component runs; and
— In-depth evaluation of components.

For deployed systems, various types of vulnerability scanners are used for networks, hosts, databases, Web applications, multi-level scanning, and automated penetration testing [Goertzel and Winograd 2011]. Statistical, data mining, or machine learning methods are also used for context-specific VA tools [Shar et al. 2013; Shin and Williams 2008].

However, previous studies have the following limitations:

— Automated vulnerability detection tools miss many vulnerabilities with high false positives because they rely on detection results and accordingly use limited information that does not reveal all details of the internal behavior [Antunes and Vieira 2015a, 2009; Vieira et al. 2009].
— Manual VA can find more vulnerabilities than automated tools although efficiency is an issue [Kupsch et al. 2010].
— Neither of manual penetration testing, static analysis, and automated penetration testing can detect all vulnerabilities. However, manual penetration testing is better at identifying design flaws; static analysis is better at finding implementation bugs; and automated penetration testing is more efficient in terms of the detection rate of vulnerabilities [Austin and Williams 2011].
— Both static analysis and automated penetration testing have a large volume of false-positives; but the former can find more SQL injection vulnerabilities [Antunes and Vieira 2015b] and more vulnerabilities in open source blogging applications [Scandariato et al. 2013].
— Automated web vulnerability scanners do not provide complete coverage over system resources (e.g., application components behind Flash applications or JavsScript-generated links) [Doupé et al. 2010].

The difference of the effectiveness measured by different studies derives from different settings or different VA tools used by each study. However, many studies agree that static analysis tools (i.e., VA analysis under non-runtime environments with or without human intervention) have performed better than automated analysis [Antunes and Vieira 2015a] because the automation may not reflect vulnerabilities derived from diverse characteristics of a system.

We address a set of more systematic metrics to evaluate VA tools. Although many of the metrics discussed below are widely used [Antunes and Vieira 2015b,a], some of the metrics are newly introduced in this work.

Let V denote the number of ground truth vulnerabilities and $\neg V$ denote the number of ground truth non-vulnerabilities with respect to a given system or set of systems. The result of a VA tool can generate: True-Positives (TP), False-Positives (FP), True-Negatives (TN), or False-Negatives (FN).

— *True-Positive Rate* (TPR) is the rate of detecting a vulnerability correctly over total vulnerabilities detected, and denoted by:

$$TPR = \frac{TP}{V} = \frac{TP}{TP + FN} \tag{3}$$

— *False-Negative Rate* (FNR) is the rate of detecting a vulnerability incorrectly over total vulnerabilities detected, and denoted by:

$$FNR = \frac{FN}{V} = \frac{FN}{TP + FN} \tag{4}$$

— *True-Negative Rate* (TNR) is the rate of detecting a non-vulnerability correctly over total non-vulnerabilities detected, and denoted by:

$$TNR = \frac{TNR}{\neg V} = \frac{TN}{FP + TN} \tag{5}$$

— *False-Positive Rate* (FPR) is the rate of detecting a non-vulnerability incorrectly over total non-vulnerabilities detected, and denoted by:

$$FPR = \frac{FP}{\neg V} = \frac{FP}{FP + TN} \tag{6}$$

where $TPR + FNR = TNR + FPR = 1$.

— *Accuracy* ($\mathcal{A}$) is the rate of detecting both vulnerabilities and non-vulnerabilities correctly over all assessments performed and denoted by:

$$\mathcal{A} = \frac{TP + TN}{TP + FN + FP + TN} \tag{7}$$

— *Precision* ($\mathcal{P}$) is the rate of detecting a vulnerability correctly over total actual vulnerabilities and denoted by:

$$\mathcal{P} = \frac{TP}{TP + FP} \tag{8}$$

This is also called *Bayesian detection rate*.

— *Recall* ($\mathcal{R}$) is the ratio between the detected true vulnerabilities (i.e., true positives) and the total number of true vulnerabilities and denoted by:

$$\mathcal{R} = \frac{TP}{TP + FN} \tag{9}$$

This metric is also called *sensitivity*. In Eq. (9), we assume that $TP + FN$ represents all relevant information where TP is relevant information over retrieved information. In this context, this metric is the same as TPR. This metric may be affected by the bias caused by prevalence and skew [Powers 2011].

— *F-measure* is a single-metric indicator of effectiveness in that a high F-Measure implies a highly effective VA tool. Assuming the precision ($\mathcal{P}$) and recall ($\mathcal{R}$) have equal weight, it is defined by:

$$F\text{-}Measure = \frac{2 \times \mathcal{P} \times \mathcal{R}}{\mathcal{P} + \mathcal{R}} = \frac{2 \times TP}{2 \times TP + FP + FN} \tag{10}$$

— *Receiver Operating Characteristic* (ROC) is to describe trade-offs between a true-positive rate and a false-positive rate via the dependence of the true-positive rate on the false-positive rate. ROC is shown with True Positive Rate (TPR) for x-axis and with False Positive Rate (FPR) for y-axis. This does not consider a base rate of vulnerabilities, and may mislead when the base rate of vulnerability is very small, known as the *base-rate fallacy* [Axelsson 2009].

— *Vulnerability Detection Operating Characteristic* (VDOC) is to consider the dependence of the True Positive Rate (TPR) on Precision ($\mathcal{P}$) accommodating the base rate $\frac{V}{V + \neg V}$. This way remedies the effect of the *base-rate fallacy*, allowing fair comparison of VA tools with different base rates. VDOC is shown with TPR for x-axis and $\mathcal{P}$ for y-axis.

— *Relative Vulnerability Detection Power* (RVDP) is inspired by a *relative strength* metric in IDS context [Boggs, Du, and Stolfo Boggs et al.; Boggs and Stolfo 2011]. A VA tool, denoted by d, does not offer any extra detection power if it cannot detect any vulnerability beyond what the current set of VA tools is able to detect, denoted by $\mathcal{D}'$. Given V is the total number of vulnerabilities, $\mathcal{D}' = \{d_1, \ldots, d_n\}$ is the set of n vulnerability detection tools, and X_d be the set of vulnerabilities detected by $d \in \mathcal{D}'$. RVDP is defined by:

$$RVDP(d', \mathcal{D}', \mathcal{D}) = \frac{|X_{d'} - \cup_{d \in \mathcal{D}} X_d|}{V} \tag{11}$$

where $d' \in \mathcal{D}'$ for $\mathcal{D} \subset \mathcal{D}'$.

— *Collective Vulnerability Detection Power* (CVDP) is also inspired by an IDS countermeasure [Boggs, Du, and Stolfo Boggs et al.; Boggs and Stolfo 2011; Mohaisen and Alrawi 2014; Morales et al. 2012]. Given V is the total number of ground truth vulnerabilities, let $\mathcal{D}' = \{d_1, \ldots, d_n\}$ be a set of VA tools, and X_d be the set of vulnerabilities detected by a VA tool $d \in \mathcal{D}'$. CVDP of a set of VA tools is denoted by:

$$CVDP(\mathcal{D}') = \frac{|\cup_{d \in \mathcal{D}'} X_d|}{V}. \tag{12}$$

where $\mathcal{D} \subseteq \mathcal{D}'$.

— *Coverage* ($\mathcal{C}$) measures the system components analyzed by a VA tool, including the physical objects (e.g., hardware, networks), cyber objects (e.g., software, data), human/social objects (e.g., users, developers, management), and enabling infrastructure (e.g., buildings, power, air).

Since a system security level changes over time as new attacks arrive or the system evolves with more sophisticated operations to perform upon new interactions between system components, new vulnerabilities can be identified. In order to capture dynamics of the vulnerabilities, we propose a dynamic VA concept as follows. Given a vector of metrics that evaluate the quality of VA tools, denoted by $\mathbf{M}(t)$ at

time t, a system can periodically re-assess system vulnerabilities. Note that $t = 0$ may correspond to the system testing time before the system deployment. Capturing historical evaluation patterns is critical to mitigating the impact of the 'base rate fallacy' [Axelsson 2009]. We summarize the above metrics used to measure vulnerabilities by VA tools in Table 1 of the appendix.

6.2. Metrics of Risk Assessment Tools

Cyber Risk Assessment (CRA) is a field that has yet to be investigated systematically. Traditional risk analysis is not adequate for security analysis in complex systems, which exhibit emergent properties [Park et al. 2013; Xu 2014c]. For example, traditional risk analysis has identified hazards. However, this is not enough for security analysis because hazards are largely unknown in complex systems and threats can emerge as the interactions between system components evolve [Park et al. 2013]. In general, security in a complex engineering system is a dynamic, emergent property that can be only observed by considering concrete attack scenarios [Haimes 2009a].

In the framework of Probability Risk Analysis (PRA) [Jensen 2002], which has been widely used, the risk metric is defined as:

$$risk = threat \times vulnerabilitiy \times consequence, \tag{13}$$

where both $threat$ and $vulnerability$ are probabilities estimated by subject experts, and $consequence$ is often measured in terms of cost incurred by the threat. This approach considers probable future scenarios with respective probabilities, estimates the consequences incurred in the scenarios, and measures risk as the expected loss or other metrics.

The PRA approach was advocated in the Department of Homeland Security's 2009 National Infrastructure Protection Plan. However, at the same time, it has been criticized by the National Research Council when applied to the threat model of intelligent, goal-oriented terrorist attacks [Concil 2010]. Researchers also have criticized the PRA approach in terms of how probabilistic assessment of terrorist attacks can be misleading [Brown and Cox 2011a,b; Cox et al. 2005; Cox 2009]. One particular weakness of the PRA approach is that it can accommodate neither the correlation or dependence between the probabilities of attack events nor the interdependency of incurred non-additivitive loss (e.g., the loss may be amplified by the effect of cascading failures). This implies that risk priority scoring systems, despite their wide use, are not appropriate for dealing with the correlated or dependent threats in the real world [Cox 2009].

As discussed earlier in Section 2.2.2, Cybersecurity Dynamics is a promising new approach for modeling and analyzing security as well as risk from a holistic perspective [Xu 2014a]. This approach is suitable for analyzing cybersecurity risk bacause of its predictive power in terms of the evolution of cyber attacks [Chen et al. 2015; Peng et al. 2016; Xu et al. 2017; Zhan et al. 2013, 2014, 2015] and the evolution trajectory of the global or network-wide cybersecurity state [Da, Xu, and Xu Da et al.; Han et al. 2014; Li et al. 2011; Lu et al. 2013; Xu and Xu 2012; Xu et al. 2015; Xu 2014a,c; Xu et al. 2012a,b, 2014, 2015; Zheng et al. 2015]. Nevertheless, there are several fundamental technical barriers that are yet to be tackled adequately, and these barriers appear to be inherent to the problem, implying that they would be encountered regardless of the specific modeling and/or analysis methods [Xu 2014a].

In order to fairly compare RA tools which have not been addressed much in the literature, we propose the metrics to characterize the effectiveness of RA tools, which is summarized in Table 2 of the appendix.

6.3. Metrics of Red Teaming

VA aims to identify potential vulnerabilities. Penetration Testing (PT) goes a step further than VA by conducting controlled exploitation of the vulnerabilities to determine their risk or consequences and approaching target systems in a similar manner as the attackers. Understanding the exploitability of potential vulnerabilities provides a substantial extra value to the defenders [Hayes 2016; RedTeams 2013].

PT focuses on discovering known, but unpatched, vulnerabilities, rather than identifying zero-day vulnerabilities. Red Team (RT) goes a step even further than PT because RT is to improve security of enterprises, such as effectiveness of existing defense and training for better defense practitions [Hayes 2016; RedTeams 2013]. RT is more focused on testing an enterprise's cyber detection and response capabilities by compromising the target systems stealthily via the exploitation of one or multiple vulnerabilities, rather than detecting as many vulnerabilities as possible. Thus, RT uses a collective set of attack patterns embracing diverse attacks / threats in hardware, software, network, human factors, and physical environments.

We define three categories of metrics to measure the quality of RT as follows: 'team competency,' 'exercise scope,' and 'test results from RT exercise.'

6.3.1. **Team Competency**. Team competence is assessed by a team's *expertise*, *techniques*, and/or *tools* required to achieve a mission aiming to exploit system vulnerabilities. We summarize how the team competence is measured in Table 3 of the appendix. It is possible to extend this metric to the more detailed classification of RT purposes, such as *understanding* the adversarial behavior, *anticipating* possible attacks, *testing* an organization's countermeasures, *reporting* recommendations to enhance an organization's security [Brangetto et al. 2015].

6.3.2. **Exercise Scope**. Exercise scope metric identifies the scope of the system under investigation in terms of the covered features or functionalities, security requirements vs. security enforcing functions included (i.e., Common Criteria Target of Evaluation [Mellado et al. 2007] or equivalent). It does not include the list of out-of-scope systems, the list the subsystems / functionality explicitly excluded from the investigation such as the areas of the system under investigation and those excluded. This class of metrics includes:

— *Lifecycle description* is the phase of a system lifecycle at which an RT is conducted, such as system test phase or operational phase;
— *System description* describes the tested components at the policy-level, architecture-level, service-level, and mechanism-level; and
— *Defense description* describes the defense tools deployed in a system under testing. This includes the following: (1) preventive defense mechanisms such as access control and encryption; (2) reactive defense mechanisms such as intrusion detection and anti-malware tools; and (3) proactive defense mechanisms such as moving-target defense tools.

6.3.3. **Test Results from RT**. This can be reported based on *successful attacks*, *defense capabilities*, and *RT outcome*. We summarize the three categories of the test results from RT based on what outcomes are measured and how suggested metrics can be measured in Table 4 of the appendix.

Based on the three factors we have discussed, including team competency, exercise scope, and test results from RT exercise, we propose a *red team resistance metric*, R_s by:

$$R_s = f(C_T, S_E, R_T) \tag{14}$$

where C_T is a team's competency, S_E is the exercise scope (i.e., coverage) of RT exercise conducted, and R_T is the test results from the RT exercise. In practice, $f(\cdot)$ can be learned from datasets describing the three components and the corresponding team resistance.

7. DISCUSSION

In this section, we discuss (1) how threats, assessment tools (i.e., VA, RA, PT, and RT), and system metrics (i.e., STRAM metrics in this work) are related to each other; and (2) what are the limitations of the currently existing system metrics and measurements methods.

7.1. Relationships between Threats, Assessment Tools, and System Metrics

In the preceding sections, we proposed a system metric framework, namely STRAM, based on ontologies of each submetric and discussed metrics of assessment tools including VA, RA, and RT. In this section, we discuss how system threats and vulnerabilities interact with the metrics of assessment tools which can affect the trustworthiness of a system.

Fig. 10 shows the relationships between system vulnerabilities (or faults) and assessment tools (i.e., VA, RA, and RT) where a system is vulnerable to attacks and protected by some defense mechanisms. System vulnerabilities (or faults) can be *identified by* VA (or PT), and the identified vulnerabilities are called 'known vulnerabilities.' RA also estimates the degree of risk based on the identified vulnerabilities, threats (i.e., faults and attacks), and its expected impact (i.e., risk considering the consequence) on the system security and performance. The known vulnerabilities are *prevented or mitigated* by various types of defense mechanisms which are also *affected by* attack patterns (e.g., power, strength, tactics, intentions). System vulnerabilities (or faults) can be *exploited by* attackers, who aim to penetrate systems, in order to breach system security mechanisms.

A system's defense mechanisms can *prevent, detect, or respond* (i.e., mitigate or block attacks) to potential or detected attacks. Based on the vulnerabilities identified by VA (or PT), the system is equipped with defense mechanisms to deal with the vulnerabilities. The defense mechanisms can be *tested by* RT which can identify new, unknown vulnerabilities and *update* the extent of system vulnerabilities. Risk can be estimated based on system vulnerabilities, threats, and consequence (i.e., an attack's impact). This implies that the quality of VA (or PT) tools *affect* the accuracy of risk estimated by RA.

From the point of view of vulnerability detection, the quality of a system can be mainly *measured by* the extent of vulnerabilities detected by assessment tools including VA (or PT), RA, and RT. Therefore, the

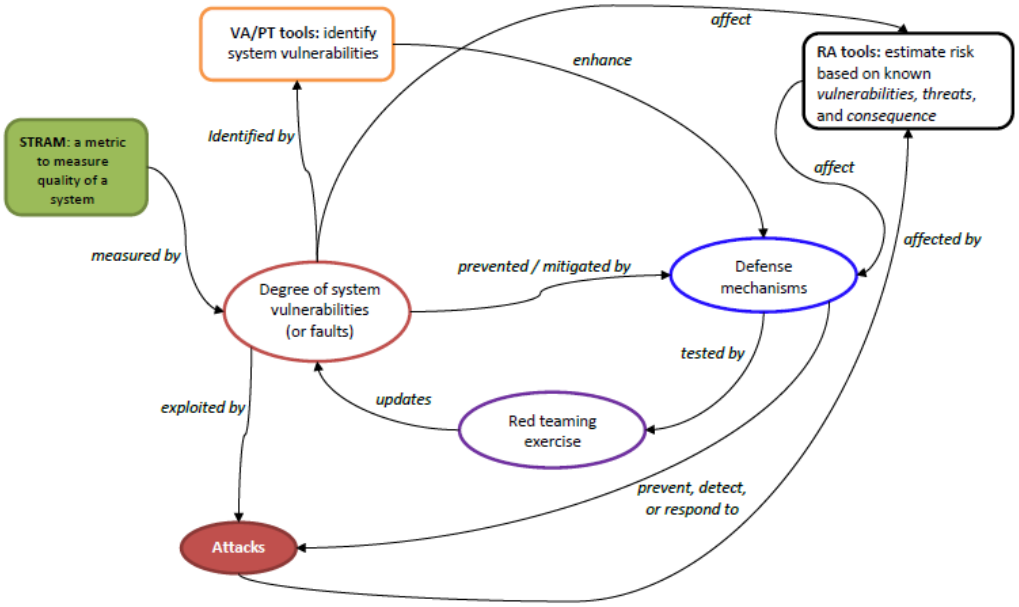

Fig. 10. Relationships between system components, assessment tools, and system metrics.

quality of assessment tools is critical to the validity of the estimated quality of the system as determined by the system trustworthiness metric, STRAM.

7.2. Limitations of Current System Metrics and Measurements

Based on the existing systems metric research literature, we identify the following hurdles:

7.2.1. **Lack of clear definitions of metric attributes**. Metric attributes can be defined differently depending on a specific domain. For example, the definition of risk (i.e., the likelihood that an attacker can exploit system vulnerabilities) in the computer system security domain is quite different from that of risk in a general sense (i.e., free from hazards). In addition, there is lack of agreement on many commonly used key metrics terms such as availability, reliability, survivability, or trustworthiness. At least we should identify a minimum set of key metrics that allows maximizing efficiency of measurements while meeting a sufficient level of expressivity to cover the multidimensional quality of a system.

7.2.2. **Lack of systematic understanding of metrics**. It is not clear what levels of abstractions are desirable for an ideal metric framework. For example, in terms of the cyber infrastructure, should we abstract a device / computer, a software component, or a data item as an abstract entity? Existing studies on resilience, especially those which conduct topological analyses, often treat a device as an abstract entity (e.g., a node in a graph). This coarse-grained granularity, while enabling analytic treatments, often oversimplifies interdependent infrastructures. On the other hand, infrastructure security and resilience are emergent properties [Pfleeger and Cunningham 2010; Xu 2014c]. This implies that a *compositional* approach to measure resilience is not possible because (1) the resilience of interdependent infrastructures cannot be derived from the resilience of the individual infrastructures; and (2) the resilience of an infrastructure cannot be derived from the resilience of its components.

7.2.3. **Lack of criteria to measure the validity of metric frameworks**. Systems metric framework is a method to measure the quality of a system. However, it is not clear to say whether a used metric framework is valid by meeting the four key factors of objectivity, efficiency, controllability, and learnability, as discussed in Section 2.1. Ultimately it is a research challenge to determine what key metric attributes are associated with two goals: expressivity (i.e., how many different aspects of system attributes should be considered) and efficiency (i.e., how efficiently each metric attribute can be obtained without much duplicated measurements). As discussed in Section 2.1, a number of metrics are used interchangeably in the literature with very little distinction (e.g., robustness vs. reliability, confidentiality vs. secrecy, maintainability vs. recoverability). If we consider all possible granulated metric attributes, it will meet high expressivity. However, this may introduce a high computational complexity in which many metrics mea-

sure the same quality aspects of a system. Making a good balance between these two conflicting goals is not a trivial task.

7.2.4. **Lack of datasets for metrics validation**. The lack of datasets is a well recognized barrier. This barrier not only prevents researchers from validating their metrics, but also prevents researchers from drawing insights from data and then using these insights to guide the definition of actionable, feasible, or analytic metrics. Indeed, it is debated that modeling natural disasters has been possible because of the availability of relevant datasets, but this is not true for malicious attacks [Santos et al. 2007].

7.2.5. **Lack of proper uncertainty handling**. Improper uncertainty handling is observed in two inherent uncertainties due to (1) the distribution of a random variable; and (2) measurement methods. These two types of uncertainties are often entangled together, but little work deals with these uncertainties properly.

7.2.6. **Lack of system-level holistic, dynamic trustworthiness metrics**. The metric framework of computer-based systems should embrace system components, hardware, software, networks, human factors, and physical environments, as discussed earlier. However, metrics to measure the multidimensional quality of the system have not been addressed in the literature. To add effort to improve this shortfall, the STRAM framework takes a holistic view for exploring trustworthiness metrics.

8. CONCLUSION

8.1. Summary

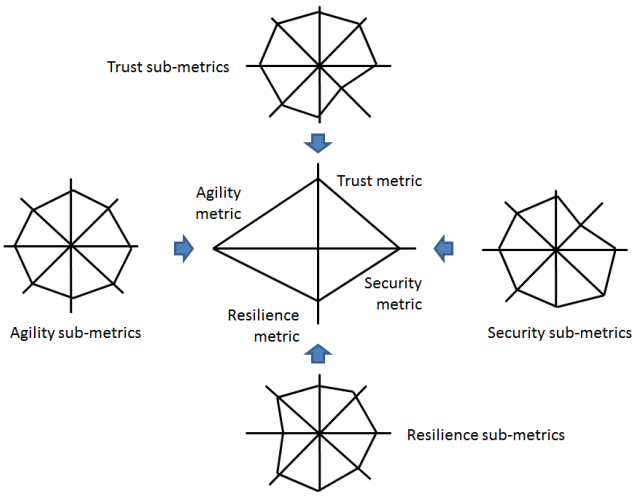

Fig. 11. Conceptual illustration of the aggregation of four submetrics in STRAM where each submetric has multiple attributes.

In this work, we conducted a systematic survey on the metrics, measurements, and metric ontologies (Sections 2 and 4), key properties of a system (Section 3), and proposed a system-level trustworthiness metric framework, namely STRAM considering Security, Trust, Resilience, and Agility. In addition, we proposed submetric ontologies under STRAM and showed a comprehensive ontology corresponding to the proposed metric framework using the Protégé [Stanford Center for Biomedical Informatics Research 2015] ontology methodology (Section 5).

We also conducted a systematic survey and discussion on metrics of assessment tools including Vulnerability Assessment, Risk Assessment, and Red Teaming. We proposed key metrics for each assessment tool (Section 6). Further, we discussed how system components (i.e., threats, vulnerability, attack, defense) are related to the assessment tools and the STRAM metric framework and the interplay between them (Section 7.1). In addition, we identified the limitations of the current system metrics research (Section 7.2).

To the best of our knowledge, this work adds value to the systems metric research community by considering the multidimensional nature of a system based on the interactions between system components and assessment tools.

8.2. Future Research Directions

We suggest the following future research directions to address the limitations of the current metrics and measurement research as discussed in Section 7.2.

— ***More comprehensive vulnerability assessment***. Is there a way of statistically assessing the degree of unknown vulnerabilities in a system based on historical records? The example records may include evaluation reports of development phases, the system type, the programming language used, the experience of developers, the type and frequency of vulnerabilities in similar existing systems, and the system complexity. This approach may guide the identification of unknown vulnerabilities which are not easily revealed through scoped and targeted threat models. This is important because, for example, a recent study [Li et al. 2016] shows that software vulnerabilities are prevalent because of many forms of code reuse, which are difficult to track in practice.

— ***How to aggregate submetrics into a higher level metric***. Well-accepted metrics are often geared towards individual attributes of building-blocks mechanisms (e.g., false-positive rate and false-negative rate of a vulnerability assessment or attack detection mechanism). However, for the purpose of decision-making in the real world, practitioners often need quantitative measurements for higher level metrics, such as security, trust, resilience, and agility. The state-of-the-art measurement method for these higher level metrics is to use the aggregation method described in Fig. 11, which is largely qualitative but can be quantified based on an average or weighted estimation. However, the average or weighted-average can often be misleading [Savage 2009], and accordingly leads more research needs to be conducted to develop appropriate aggregation functions preserving inherent concepts and relationships between the lower-level submetrics while minimizing the redundant, duplicate measurements of highly similar metric attributes. The most challenging task is to determine dependencies between metrics based on certain criteria to maintain the validity of metrics in order to make effective and efficient balances to cover the most aspects of system quality while not too much overlapping the measurement coverage between metrics.
— ***Quantitative calculation of the four submetrics***. Although this work proposed a system-level trustworthiness metric framework, we have not discussed the details of the computation of quantitative metrics representing the four key submetrics, security, trust, resilience, and agility. In our future work, we aim to develop metrics that can measure the system quality measuring the four metrics. In particular, we will be more interested in developing resilience and agility metrics that can capture more dynamic aspects of system quality.
— ***Determining key system metrics***. Key system metrics can be different depending on system contexts and requirements. For example, a system running on a mobile, wireless network needs more lightweight solutions with cost-effective mechanisms for measurement purposes. On the other hand, if a system has a large data volume which is maintained by a centralized data center and needs to provide optimal solutions to users, service reliability with high accuracy can be a top priority metric. Therefore, an appropriate selection of key system metrics is an open research question that can be affected by conflicting system goals. Moreover, the key system metrics for military systems may be different from the key system metrics for civilian systems. The former may emphasize more on successful mission completion. Identifying key system metrics remains a challenging open research question.
— ***Repeatability of experimental measurements***. For each key system metric to characterize the trustworthiness of systems, we need to consider a measurement procedure to capture each metric. It is critical to validating and testing these measurement procedures under dynamic, hostile, and/or high-temp network environments, such as military tactical networks. Even if a metric cannot be measured with absolute objectivity, we should investigate protocols or procedures by which metrics can be measured with comparable precision, acceptable complexity, and appropriate level of granularity. Moreover, the measurement procedures must be repeatable.
— ***System-level holistic, dynamic metric framework***. Since the quality of a single system is associated with multiple dimensions of system components, a demanding metric framework should take a system-level holistic perspective as follows: (1) diverse metric attributes should be considered embracing the quality of all dimensions of a system; (2) estimation of dynamic system states should be captured by expanding the metric dimensions from cybersecurity dynamics [Xu 2014b] to other metrics (e.g., trust, resilience, or agility, in addition to security); and (3) dynamic states of a system's situation awareness and understanding should be estimated and reflected for critical system decision making process (e.g., attack detection or defense mechanism selection). The proposed STRAM framework can be seen as a first step towards this ultimate goal.

In the past, the trustworthiness of a system has often been measured by its ability to be evaluated, usually against some known criteria (e.g., common criteria such as Evaluation Assurance Level, or EAL, or Protection Level) albeit under a system-defined Target of Evaluation (TOE). Having an objective indicator of the trustworthiness of a system based on the proposed STRAM framework will allow a broader range of security analyses to be performed. Examples may include the ability to: compare systems using a consistent set of metrics; perform a thorough gap analysis on a system or product to determine which aspects need more attention to ensure trustworthiness; identify important common metrics of the most trustworthy systems; define mandatory sets of metrics for differing levels of trustworthiness and differing dynamic contexts; identify the weighted importance of different metrics and measurements; and report on the different axes of security posture, including physical, logical, and human factors.

Over time through the application of these use cases, the STRAM framework can be tailored for different contexts, and the specific metrics and measurements can be refined and even weighted. The use cases also serve to develop a consistent mapping of products or systems to the STRAM framework, this

mapping is not only obtaining the appropriate measurements, but instrumenting systems such that they are amenable to metric analysis.

Acknowledgement

This research was in part supported by the US Department of Defense (DoD) through the office of the Assistant Secretary of Defense for Research and Engineering (ASD (R&E)) and ARO Grant #W911NF-17-1-0566. The views and opinions of the author(s) do not reflect those of the US DoD, ASD (R&E), or US Army, Navy, or Airforce. In addition, they do not represent those of the UK Ministry of Defence as well as those of the Australia Ministry of Defence.

REFERENCES

W. Neil Adger. 2000. Social and ecological resilience: are they related? *Progress in Human Geography* 24, 3 (2000), 347–364.

W. Richards Adrion, Martha A. Branstad, and John C. Cherniavsky. 1982. Validation, Verification, and Testing of Computer Software. *ACM Comput. Surv.* 14, 2 (June 1982), 159–192.

M. H. Al-Kuwaiti, N. Kyriakopoulos, and S. Hussein. 2008. Towards a Standardized Terminology for Network Performance. *IEEE Transactions on Reliability* 57, 2 (June 2008), 267–271.

D. S. Alberts. 2007. Agility, Focus, and Convergence: The Future of Command and Control. *The International C2 Journal* 1, 1 (2007), 1–30.

David S. Alberts. 2011. *The Agility Advantage: A Survival Guide for Complex Enterprises and Endeavors*. CCRP.

David S. Alberts. 2014. Agility Quotient (AQ). In *Proceedings of the 19th International Command anc Control Research and Technology Symposium (ICCRTS'14)*.

D. L. Alderson, G. G. Brown, W. M. Carlyle, and L. A. Cox. 2013. Sometimes there is no most-vital arc: assessing and improving the operational resilience of systems. *Military Oper. Res.* 18, 1 (2013), 21–37.

P. J. Antsaklis, K. M. Passino, and S. J. Wang. 1989. Towards intelligent autonomous control systems: Architecture and fundamental issues. *Journal of Intelligent and Robotic Systems* 1, 4 (01 Dec 1989), 315–342.

Nuno Antunes and Marco Vieira. 2009. Comparing the Effectiveness of Penetration Testing and Static Code Analysis on the Detection of SQL Injection Vulnerabilities in Web Services. In *Proceedings of the 2009 15th IEEE Pacific Rim International Symposium on Dependable Computing (PRDC '09)*. 301–306.

Nuno Antunes and Marco Vieira. 2015a. Assessing and Comparing Vulnerability Detection Tools for Web Services: Benchmarking Approach and Examples. *IEEE Transactions on Services Computing* 8, 2 (2015), 269–283.

N. Antunes and M. Vieira. 2015b. On the Metrics for Benchmarking Vulnerability Detection Tools. In *2015 45th Annual IEEE/IFIP International Conference on Dependable Systems and Networks*. 505–516.

Andrew Austin and Laurie Williams. 2011. One Technique is Not Enough: A Comparison of Vulnerability Discovery Techniques. In *Proceedings of the 2011 International Symposium on Empirical Software Engineering and Measurement (ESEM '11)*. 97–106.

A. Avizienis, J. C. Laprie, B. Randell, and C. Landwehr. 2004. Basic Concepts and Taxonomy of Dependable and Secure Computing. *IEEE Transactions on Dependable and Secure Computing* 1, 1 (Jan–Mar. 2004), 11–33.

Stefan Axelsson. 2009. The Base-rate Fallacy and Its Implications for the Difficulty of Intrusion Detection. In *Proc. ACM Conference on Computer and Communications Security (CCS'09)*. 1–7.

Nicholas J. Bahr. 2014. *System Safety Engineering and Risk Assessment: A Practical Approach*. CRC Press, Chapter Risk Assessment.

Albert-L/'aszló Barabási. 2016. *Network Science*. Cambridge University Press, Chapter Network robustness.

Carlos Blanco, Joaquín Lasheras, Eduardo Fernández-Medina, Rafael Valencia-García, and Ambrosio Toval. 2011. Basis for an integrated security ontology according to a systematic review of existing proposals. *Computer Standards & Interfaces* 33, 4 (June 2011), 372–388.

E. Blasch. 2014. Trust metrics in information fusion. In *Proc. SPIE 9119*. Machine Intelligence and Bio-inspired Computation: Theory and Applications VIII, 91190L.

Nathaniel Boggs, Senyao Du, and SalvatoreJ. Stolfo. Measuring Drive-by Download Defense in Depth. In *Proc. RAID'14*. 172–191.

Nathaniel Gordon Boggs and Salvatore Stolfo. 2011. ALDR: A New Metric for Measuring Effective Layering of Defenses. *Fifth Layered Assurance Workshop (LAW'11)* (2011).

R. Böhme and F. C. Freiling. 2008. *Dependability Metrics*. Vol. 4909. Springer-Verlag Lecture Notes in Computer Science, Chapter On Metrics and Measurements, 7–13.

W.N. Borst. 1997. *Construction of Engineering Ontologies*. Centre for Telematica and Information Technology, University of Tweente, Eschede.

Pascal Brangetto, Emin Caliskan, and Henry Roigas. 2015. NATO CCDCOE Cyber Red Teaming: Organisational, technical and legal implications in a military context (2015). (2015). https://ccdcoe.org/sites/default/files/multimedia/pdf/Cyber_Red_Team.pdf

Nick Brooks. 2003. *Vulnerability, risk and adaptation: A conceptual framework*. Technical Report 38. Tyndall Centre for Climate Change Research. Tyndall Centre Working Paper.

Gerald Brown, Matthew Carlyle, Javier Salmerón, and Kevin Wood. 2006. Defending Critical Infrastructure. *Interfaces* 36, 6 (2006), 530–544.

Gerald G. Brown and Louis Anthony (Tony) Cox, Jr. 2011a. How Probabilistic Risk Assessment Can Mislead Terrorism Risk Analysts. *Risk Analysis* 31, 2 (2011), 196–204.

Gerald G. Brown and Louis Anthony (Tony) Cox, Jr. 2011b. Making Terrorism Risk Analysis Less Harmful and More Useful: Another Try. *Risk Analysis* 31, 2 (2011), 193–195.

A. Burns, J. McDermid, and J. Dobson. 1992. On the Meaning of Safety and Security. *Comput. J.* 35, 1 (1992).

Lawrence A. ; Loeb Martin P. ; Zhou Lei Campbell, Katherine; Gordon. 2003. The economic cost of publicly announced information security breaches: empirical evidence from the stock market. *Journal of Computer Security* 11, 3 (2003), 431–448.

K. Chan, J.H. Cho, T. Trout, J. Wampler, A. Toth, and B. Rivera. 2015. Trustd: Trust Daemon Experimental Testbed for Network Emulation. In *IEEE Military Communications Conference*.

E. Chang, T. S. Dillon, and F. Hussain. 2007. Trust Ontologies for E-Service Environments. *International Journal of Intelligent Systems* 22 (2007), 519–545.

Huashan Chen, Jin-Hee Cho, and Shouhuai Xu. 2018. Quantifying the Security Effectiveness of Firewalls and DMZs. In *Proceedings of the 5th Annual Symposium and Bootcamp on Hot Topics in the Science of Security (HoTSoS '18)*. Article 9, 9:1–9:11 pages.

Yu-Zhong Chen, Zi-Gang Huang, Shouhuai Xu, and Ying-Cheng Lai. 2015. Spatiotemporal patterns and predictability of cyberattacks. *PLoS One* 10, 5 (05 2015), e0124472.

Brian Chess and Jacob West. 2007. *Secure Programming with Static Analysis* (first ed.). Addison-Wesley Professional.

J.H. Cho, A. Swami, and I.R. Chen. 2011. A survey of trust management in mobile ad hoc networks. *IEEE Communications Surveys and Tutorials* 13, 4 (2011), 562–583.

Jin-Hee Cho. 2015. Tradeoffs between Trust and Survivability for Mission Effectiveness in Tactical Networks. *IEEE Transactions on Cybernetics* 45, 4 (2015).

Jin-Hee Cho, I. Alsmadi, and Dianxiang Xu. 2016. Privacy and Social Capital in Online Social Networks. In *IEEE Global Communications Conference (GLOBECOM 2016)*. Wahington D.C., USA.

Jin-Hee Cho, Hasan Cam, and Alessandro Oltramari. 2016. Effect of Personality Traits on Trust and Risk to Phishing Vulnerability: Modeling and Analysis. In *IEEE International Multi-Disciplinary Conference on Cognitive Methods in Situation Awareness and Decision Support (CogSIMA 2016)*.

Jin-Hee Cho and Ing-Ray Chen. 2016. PROVEST: Provenance-based Trust Model for Delay Tolerant Networks. *IEEE Transactions on Dependable and Secure Computing* (2016).

P. Cholda, J. Tapolcai, T. Cinkler, K. Wajda, and A. Jajszczyk. 2009. Quality of Resilience as a Network Reliability Characterization Tool. *IEEE Networks* 23, 2 (March/April 2009), 11–19.

Jason A. Colquitt, Brent A. Scott, and Jeffery A. LePine. 2007. Trust, Trustworthiness, and Trust Propensity: A Meta-Analytic Test of Their Unique Relationships With Risk Taking and Job Performance. *Journal of Applied Psychology* 92, 4 (2007), 909–927.

K. Conboy. 2009. Agility from First Principles: Reconstructing the Concept of Agility in Information Systems Development. *Information Systems Research* 20, 3 (Sept. 2009), 329–354.

National Research Concil. 2010. *Review of the Department of Homeland Security's Approach to Risk Analysis*. The National Academies Press.

Louis Anthony (Tony) Cox, Djangir Babayev, and William Huber. 2005. Some Limitations of Qualitative Risk Rating Systems. *Risk Analysis* 25, 3 (2005), 651–662.

Louis Anthony (Tony) Cox, Jr. 2009. What's Wrong with Hazard-Ranking Systems? An Expository Note. *Risk Analysis* 29, 7 (2009), 940–948.

Gaofeng Da, Maochao Xu, and Shouhuai Xu. A new approach to modeling and analyzing security of networked systems. In *Proc. 2014 Symposium and Bootcamp on the Science of Security (HotSoS'14)*. 6.

Weiqi Dai, T. Paul Parker, Hai Jin, and Shouhuai Xu. 2012. Enhancing Data Trustworthiness via Assured Digital Signing. *IEEE Trans. Dependable Sec. Comput.* 9, 6 (2012), 838–851.

Darren Davis, Fabian Monrose, and Michael K. Reiter. 2004. On User Choice in Graphical Password Schemes. In *Proceedings of the 13th USENIX Security Symposium*. San Diego, CA.

A. H. Dekker. 2006. Measuring the Agility of Networked Military Forces. *Journal of Battlefield Technology* 9, 1 (March 2006), 1–6.

Dorothy E. Denning. 1999. *Information Warfare and Security*.

Department of Homeland Security. 2015. National Critical Infrastructure Security and Resilience Research and Development Plan. http://publish.illinois.edu/ciri-new-theme/files/2016/09/National-CISR-RD-Plan_Nov-2015.pdf. (2015).

Morton Deutsch. 1960. Trust, trustworthiness, and the F scale. 61 (08 1960), 138–40.

N. Dokoohaki and M. Matskin. 2007. Structural Determination of Ontology-Driven Trust Networks in Semantic Social Institutions and Ecosystems. In *International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies*. Papeete, France, 263–268.

Marc Donner. 2003. Towards a Security Ontology. *IEEE Security & Privacy* (2003).

Adam Doupé, Marco Cova, and Giovanni Vigna. 2010. Why Johnny Can'T Pentest: An Analysis of Black-box Web Vulnerability Scanners. In *Proceedings of the 7th International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment (DIMVA'10)*. 111–131.

Rick Dove. 2001. *Response Ability: The Language, Structure, and Culture of the Agile Enterprise*. John Wiley & Sons.

Rick Dove. 2005. Fundamental Principles for Agile Systems Engineering. In *Proc. Conference on Systems Engineering Research (CSER'05)*.

N. Edwards, Paul. 1996. *The Closed World: Computers and the Politics of Discourse in Cold War America*. Cambridge, MA: MIT Press.

ENISA. 2011. Ontology and taxonomies of resilience.

B. J. Fogg, Jonathan Marshall, Othman Laraki, Alex Osipovich, Chris Varma, Nicholas Fang, Jyoti Paul, Akshay Rangnekar, John Shon, Preeti Swani, and Marissa Treinen. 2001. What Makes Web Sites Credible?: A Report on a Large Quantitative Study. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '01)*. ACM, New York, NY, USA, 61–68.

Stephen R.; Walker Brian; Scheffer Marten; Chapin Terry; Rockstrom Johan.; Folke, Carl; Carpenter. 2010. Resilience thinking: integrating resilience, adaptability and transformability. *Ecology and Society* 15, 4 (2010).

JN Goel and BM Mehtre. 2015. Vulnerability Assessment and Penetration Testing as a Cyber Defence Technology. *Procedia Computer Science* 57 (2015), 710–715.

Karen Mercedes Goertzel and Theodore Winograd. 2011. *Information Assurance Tools Report – Vulnerability Assessment* (sixth ed.). Technical Report. IATAC.

J. Golbeck and B. Parsia. 2006. Trust network-based filtering of aggregated claims. *International Journal of Metadata, Semantics and Ontologies* 1, 1 (2006), 58–65.

Le Guan, Jingqiang Lin, Bo Luo, Jiwu Jing, and Jing Wang. 2015. Protecting Private Keys Against Memory Disclosure Attacks Using Hardware Transactional Memory. In *Proceedings of the 2015 IEEE Symposium on Security and Privacy (SP '15)*. 3–19.

N. Guarino. 1998. Formal Ontology in Information Systems. IOS Press, Amsterdam, 3–15.

Theo Haerder and Andreas Reuter. 1983. Principles of Transaction-Oriented Database Recovery. *Comput. Surveys* 15, 4 (Dec. 1983).

William J. Haga and Moshe Zviran. 1991. Question-and-answer passwords: An empirical evaluation. *Information Systems* 16, 3 (1991), 335–343.

Yacov Y. Haimes. 2009a. On the Definition of Resilience in Systems. *Risk Analysis* 29, 4 (2009), 498–501.

Y. Y. Haimes. 2009b. Perspective On the Definition of Resilience in Systems. *Risk Analysis* 29, 4 (2009), 498–501.

Yujuan Han, Wnelian Lu, and Shouhuai Xu. 2014. Characterizing the Power of Moving Target Defense via Cyber Epidemic Dynamics. In *Proc. 2014 Symposium and Bootcamp on the Science of Security (HotSoS'14)*. 10:1–10:12.

Keith Harrison and Shouhuai Xu. 2007. Protecting Cryptographic Keys from Memory Disclosure Attacks. In *The 37th Annual IEEE/IFIP International Conference on Dependable Systems and Networks, DSN 2007, 25-28 June 2007, Edinburgh, UK, Proceedings*. 137–143.

Wilhelm Hasselbring and Ralf Reussner. 2006. Toward Trustworthy Software Systems. *IEEE Computer* 39, 4 (April 2006), 91–92.

Shlomo Havlin, N. A. M. Araujo, Sergey V. Buldyrev, C. S. Dias, Roni Parshani, G. Paul, and H. Eugene Stanley. 2010. Catastrophic Cascade of Failures in Interdependent Networks. *Nature* 464 (2010), 1025 – 1028.

Kirk Hayes. 2016. Penetration Test vs. Red Team Assessment: The Age Old Debate of Pirates vs. Ninjas Continues. (June 2016). https://community.rapid7.com/community/infosec/blog/2016/06/23/penetration-testing-vs-red-teaming-the-age-old-debate-of-pirates-vs-ninja-continues

C. Holling. 1973a. Resilience and stability of ecological systems. *Annual Review of Ecology and Systematics* 4 (1973), 1–23.

C. S. Holling. 1973b. Resilience and Stability of Ecological Systems. *Annual Review of Ecology and Systematics* 4 (1973), 1–23.

Jin B. Hong and Dong Seong Kim. 2016. Assessing the Effectiveness of Moving Target Defenses Using Security Models. *IEEE Transactions on Dependable and Secure Computing* 13, 2 (March–April 2016), 163–177.

Hui-Min Huang, Elena Messina, and James S. Albus. 2004. Toward a Generic Model for Autonomy Levels for Unmanned Systems (ALFUS).

H. O. Hundley and R. H. Anderson. 1995. Emerging challenge: security and safety in cyberspace. *IEEE Technology and Society Magazine* 14, 4 (Winter 1995), 19–28.

Information Technology Laboratory, Software and Systems Division. Reitrieved on 2 Dec. 2016. Metrics and Measures. (Reitrieved on 2 Dec. 2016). https://samate.nist.gov/index.php/Metrics_and_Measures.html

ISO/IEC 1998. *9241-14 Ergonomic requirements for office work with visual display terminals (VDT)s-Part 14 Menu dialogues*. ISO/IEC. ISO/IEC 9241-14: 1998(E).

Philip J. Ivanhoe. 1998. Xin (trustworthiness). (1998).

Uwe Jensen. 2002. Probabilistic Risk Analysis: Foundations and Methods. *J. Amer. Statist. Assoc.* 97, 459 (2002), 925–925.

Rayford B. Vaughn Jr., Ronda Henning, and Ambareen Siraj. 2003. Information Assurance Measures and Metrics - State of Practice and Proposed Taxonomy. In *Proceedings of the 36th Hawaii International Conference on System Sciences*.

P. T. Kidd. 1994. *Agile Manufacturing: Forging New Frontiers*. Addison-Wesley, MA.

Anya Kim, Jim Luo, and Myong Kang. 2005. *Security Ontology for Annotating Resources*. Technical Report. Naval Research Laboratory.

James A. Kupsch, Barton P. Miller, Elisa Heymann, and Eduardo César. 2010. First Principles Vulnerability Assessment. In *Proceedings of the 2010 ACM Workshop on Cloud Computing Security Workshop (CCSW '10)*. 87–92.

Tong Li, Jennifer Horkoff, and John Mylopoulos. 2016. Holistic security requirements analysis for socio-technical systems. *Software & Systems Modeling* (2016).

X. Li, P. Parker, and S. Xu. 2011. A Stochastic Model for Quantitative Security Analysis of Networked Systems. *IEEE Transactions on Dependable and Secure Computing* 8, 1 (2011), 28–43.

Zhen Li, Deqing Zou, Shouhuai Xu, Hai Jin, Hanchao Qi, and Jie Hu. 2016. VulPecker: an automated vulnerability detection system based on code similarity analysis. In *Proceedings of the 32nd Annual Conference on Computer Security Applications, ACSAC 2016, Los Angeles, CA, USA, December 5-9, 2016*. 201–213.

Wenlian Lu, Shouhuai Xu, and Xinlei Yi. 2013. Optimizing Active Cyber Defense Dynamics. In *Proceedings of the 4th International Conference on Decision and Game Theory for Security (GameSec'13)*. 206–225.

Asad M. Madni and S. Jackson. 2009. Towards a Conceptual Framework for Resilience Engineering. *IEEE Systems Journal* 3, 2 (2009), 181–191.

Patrick McDaniel, Trent Jaeger, Thomas F. La Porta, Nicolas Papernot, Robert J. Walls, Alexander Kott, Lisa Marvel, Ananthram Swami, Prasant Mohapatra, Srikanth V. Krishnamurthy, and Iulian Neamtiu. 2014. Security and Science of Agility. In *Proceedings of the First ACM Workshop on Moving Target Defense (MTD'14)*. 13–19.

Gary McGraw. 2006. *Software Security: Building Security In*. Addison-Wesley Professional.

Daniel Mellado, Eduardo Fernández-Medina, and Mario Piattini. 2007. A common criteria based security requirements engineering process for the development of secure information systems. *Computer Standards & Interfaces* 29, 2 (2007), 244–253.

J. D. Mireles, J.H. Cho, and S. Xu. 2016. Extracting Attack Narratives from Traffic Datasets. In *International Conference on Cyber Conflict (CyCon)*.

Aziz Mohaisen and Omar Alrawi. 2014. Av-meter: An evaluation of antivirus scans and labels. In *Proc. DIMVA'2014*. 112–131.

Mohammadi et al. 2014. *Cloud Computing and Services Science*. Vol. 453. Springer International Publishing, Switzerland, Chapter Trustworthiness Attributes and Metrics for Engineering Trusted Internet-Based Software Systems, 19–35.

Jose Andre Morales, Shouhuai Xu, and Ravi Sandhu. 2012. Analyzing Malware Detection Eciency with Multiple Anti-Malware Programs. *ASE Science Journal* 1, 2 (2012), 56–66.

D.M. Nicol, William H. Sanders, and Kishor S. Trivedi. 2004. Model-Based Evaluation: From Dependability to Security. *IEEE Transactions on Dependable and Secure Computing* 1, 1 (Jan.–Mar. 2004), 48–65.

J. Nielsen. 1993. *Usability Engineering*. Academic Press, Inc., San Diego, CA, USA.

J. Park, T. P. Seager, P. S. C. Rao, M. Convertino, and I. Linkov. 2013. Integrating Risk and Resilience Approaches to Catastrophe Management in Engineering Systems. *Risk Analysis* 33, 3 (2013), 356–367.

Simon E. Parkin, Aad van Moorsel, and Robert Coles. 2009. An information security ontology incorporating human-behavioural implications. In *ACM Proceedings of the 2nd international conference on Security of information and networks (SIN)*.

Raymond Paul et al. 2008. An Ontology-Based Integrated Assessment Framework for High-Assurance Systems. In *IEEE International Conference on Semantic Computing*. 386–393.

Marcus Pendleton, Richard Garcia-Lebron, Jin-Hee Cho, and Shouhuai Xu. 2016. A Survey on Systems Security Metrics. *ACM Comput. Surv.* 49, 4 (Dec. 2016), 62:1–62:35.

Marcus Pendleton, Richard Garcia-Lebron, Jin-Hee Cho, and Shouhuai Xu. 2017. A survey on systems security metrics. *ACM Computing Surveys (CSUR)* 49, 4 (2017), 62.

Chen Peng, Maochao Xu, Shouhuai Xu, and Taizhong Hu. 2016. Modeling and predicting extreme cyber attack rates via marked point processes. *Journal of Applied Statistics* 0, 0 (2016), 1–30.

C. Perrings. 2006. Resilience and sustainable development. *Environment and Development Economics* 11, 4 (008 2006), 417–427.

Charles P. Pfleeger. 2006. *Security in Computing*. Prentice Hall, Chapter Is there a security problem in computing.

S.L. Pfleeger and R.K. Cunningham. 2010. Why Measuring Security Is Hard. *Security Privacy, IEEE* 8, 4 (July 2010), 46–54.

S. Pimm. 1984. The complexity and stability of ecosystems. *Nature* 307, 5945 (1984), 321–326.

Leo L. Pipino, Yang W. Lee, and Richard Y. Yang. 2002. Data Quality Assessment. *Communications of ACM* 45 (April 2002), 211–218.

D. M. W. Powers. 2011. Evaluation: From precision, recall and f-measure to roc., informedness, markedness & correlation. *Journal of Machine Learning Technologies* 2, 1 (2011), 37–63.

Practical Software and Systems Measurement. 2006. *Security Measurement*. Technical Report. PSM Safety & Security TWG.

A. Ramos, M. Lazar, R. H. Filho, and J. J. P. C. Rodrigues. 2017. Model-Based Quantitative Network Security Metrics: A Survey. *IEEE Communications Surveys Tutorials* 19, 4 (2017), 2704–2734.

M. Rausand and A. Hoyland. 2009. *System Reliability Theory: Models and Statistical Methods*. John Wiley & Sons.

RedTeams. March 22, 2013. Difference between Penetration Testing and Red Team Exercies. (March 22, 2013). http://redteams.net/blog/2013/difference-between-penetration-testing-and-red-team-exercises

ReliaSoft Corporation. 2003. Reliability HotWire: The eMagazine for the Reliability Professional. (April 2003). http://www.weibull.com/hotwire/issue26/relbasics26.htm

Markku Salo, Evangelos Markopoulos, Hannu Vanharanta, and Jussi Ilari Kantola. 2016. *Advances in Human Factors, Business Management, Training and Education, Part XVII*. Vol. 498. Chapter Degree of Agility with an Ontology Based Application.

Joost R. Santos, Yacov Y. Haimes, and Chenyang Lian. 2007. A Framework for Linking Cybersecurity Metrics to the Modeling of Macroeconomic Interdependencies. *Risk Analysis* 27, 5 (2007), 1283–1297. DOI: https://doi.org/10.1111/j.1539-6924.2007.00957.x

Sam L. Savage. 2009. *The Flaw of Averages: Why We Underestimate Risk in the Face of Uncertainty*. Wiley.

Riccardo Scandariato, James Walden, and Wouter Joosen. 2013. Static analysis versus penetration testing: A controlled experiment. *2013 IEEE 24th International Symposium on Software Reliability Engineering (ISSRE)* 00 (2013), 451–460.

Christian M. Schneider, Andre A. Moreira, Jose S. Andrade, Shlomo Havlin, and Hans J. Herrmann. 2011. Mitigation of malicious attacks on networks. *Proceedings of the National Academy of Sciences* 108, 10

(2011), 3838–3841.
F. Schneider (Ed.). 1999. *Trust in Cyberspace*. National Academy Press.
F. Schuster, T. Tendyck, C. Liebchen, L. Davi, A.-R. Sadeghi, and T. Holz. 2015. Counterfeit Object-oriented Programming: On the Difficulty of Preventing Code Reuse Attacks in C++ Applications. In *2015 IEEE Symposium on Security and Privacy*. 745–762.
Lwin Khin Shar, Hee Beng Kuan Tan, and Lionel C. Briand. 2013. Mining SQL Injection and Cross Site Scripting Vulnerabilities Using Hybrid Program Analysis. In *Proceedings of the 2013 International Conference on Software Engineering (ICSE '13)*. 642–651.
Bohdana Sherehiy, Waldemar Karwowski, and John K. Layer. 2007. A review of enterprise agility: Concepts, frameworks, and attributes. *International Journal of Industrial Ergonomics* 7 (2007), 445–460.
Hung-Pin Shih. 2004. An empirical study on predicting user acceptance of e-shopping on the Web. *Information & Management* 41 (2004), 351–368.
Yonghee Shin and Laurie Williams. 2008. An Empirical Model to Predict Security Vulnerabilities Using Code Complexity Metrics. In *Proceedings of the Second ACM-IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM '08)*. 315–317.
Gopalan Sivathanu, Charles P. Wright, and Erez Zadok. 2005. Ensuring Data Integrity in Storage: Techniques and Applications. In *First ACM Workshop on Storage Security and Survivability*.
R. Slayton. 2015. Measuring Risk: Computer Security Metrics, Automation, and Learning. *IEEE Annals of the History of Computing* 37, 2 (April-June 2015), 32–45.
A.C. Squicciarini, E. Bertino, E. Ferrari, and I. Ray. 2006. Achieving privacy in trust negotiations with an ontology-based approach. *IEEE Transactions on Dependable and Secure Computing* 3, 1 (Jan.–March 2006), 13–30.
Stanford Center for Biomedical Informatics Research. 2015. WebProtégé. (2015). http://protege.stanford.edu/.
Frank J Stech, Kristin E Heckman, and Blake E Strom. 2016. Integrating Cyber-D&D into Adversary Modeling for Active Cyber Defense. In *Cyber Deception*. Springer, 1–24.
Dafydd Stuttard and Marcus Pinto. 2007. *The Web Application Hacker's Handbook: Discovering and Exploiting Security Flaws*. John Wiley & Sons, Inc., New York, NY, USA.
M. Taherian, R. Jalili, and M. Amini. 2008. PTO: A Trust Ontology for Pervasive Environments. In *22nd International Conference on Advanced Information Networking and Applications - Workshops (AINAW'2008)*. Okinawa, Japan, 301–306.
The Forum of Incident Response and Security Teams. 2015. The Common Vulnerability Scoring System (CVSS). (June 2015). https://www.first.org/cvss
B. Tsoumas and D. Gritzalis. 2006. Towards an Ontology-based Security Management. In *IEEE 20th International Conference on Advanced Information Networking and Applications (AINA)*.
TTCP. 2014. The Technical Cooperation Program. (2014). http://www.acq.osd.mil/ttcp/index.html.
U.S. DoD. 2015. U.S. Department of Defense Cybersecurity Test and Evaluation Guidebook Version 1.0. (July 2015). http://www.dote.osd.mil/docs/TempGuide3/Cybersecurity_TE_Guidebook_July1_2015_v1_0.pdf
M. Vieira, N. Antunes, and H. Madeira. 2009. Using web security scanners to detect vulnerabilities in web services. In *2009 IEEE/IFIP International Conference on Dependable Systems Networks*. 566–571.
L. Viljanen. 2005. *Trust, Privacy, and Security in Digital Business*. Vol. 3592. Springer-Verlag Berlin Heidelberg, Lecture Notes in Computer Science, Chapter Towards an Ontology of Trust, 175–184.
Panagiotis T. Vlacheas et al. 2011. *Ontology and taxonomies of resilience*. Technical Report. European Network and Information Security Agency.
B. Walker, C.S. Hollings, S. R. Carpenter, and A. Kinzig. 2004. Resilience, Adaptability and Transformability in Social-Ecological Systems. *Ecology and Society* 9, 2 (2004), Article 5.
Jiang Wang, Angelos Stavrou, and Anup Ghosh. 2010. *HyperCheck: A Hardware-Assisted Integrity Monitor*. Springer Berlin Heidelberg, Berlin, Heidelberg, 158–177.
Xinlei Wang, Jin-Hee Cho, Kevin Chan, MoonJeong Chang, Ananthram Swami, and Prasant Mohapatra. 2013. Trust and independence aware decision fusion in distributed networks. In *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2013 IEEE International Conference on*. IEEE, 481–486.
Steve H. Weingart. 2000. *Lecture Note in Computer Science*. Vol. 1965. Springer-Verlag Berlin Heidelberg, Chapter Physical Security Devices for Computer Subsystems: A Survey of Attacks and Defenses.
B. J. Wood and R. A. Duggan. 2000. Red Teaming of advanced information assurance concepts. In *Proc. of DARPA Information Survivability Conference and Exposition*, Vol. 2. 112–118.

David D. Woods. 1996. *Automation and Human Performance: Theory and Applications*. CRC Press, Chapter Decomposing Automation: Apparent Simplicity, Real Complexity.
David D. Woods and Erik Hollnagel. 2006. *Resilience Engineering: Concepts and Precepts*. CRC Press.
Maochao Xu, Gaofeng Da, and Shouhuai Xu. 2015. Cyber Epidemic Models with Dependences. *Internet Mathematics* 11, 1 (2015), 62–92.
Maochao Xu, Lei Hua, and Shouhuai Xu. 2017. A Vine Copula Model for Predicting the Effectiveness of Cyber Defense Early-Warning. *Technometrics* 0, ja (2017), 0–0. DOI:https://doi.org/10.1080/00401706.2016.1256841 arXiv:http://dx.doi.org/10.1080/00401706.2016.1256841
Maochao Xu and Shouhuai Xu. 2012. An Extended Stochastic Model for Quantitative Security Analysis of Networked Systems. *Internet Mathematics* 8, 3 (2012), 288–320.
Shouhuai Xu. 2007. On the security of group communication schemes. *Journal of Computer Security* 15, 1 (2007), 129–169.
Shouhuai Xu. 2010. Toward a theoretical framework for trustworthy cyber sensing. In *Proc. of SPIE Defense, Security, and Sensing*.
Shouhuai Xu. 2014a. Cybersecurity Dynamics. In *Proc. Symposium and Bootcamp on the Science of Security (HotSoS'14)*. 14:1–14:2.
Shouhuai Xu. 2014b. Cybersecurity Dynamics: With Application to Formulating a Cyber Defense C2 Framework. Presentation at Workshop on Cyber Security: From Tactics to Strategy and Back. (2014).
Shouhuai Xu. 2014c. Emergent Behavior in Cybersecurity. In *Proceedings of the 2014 Symposium and Bootcamp on the Science of Security (HotSoS'14)*. 13:1–13:2.
Shouhuai Xu, Wenlian Lu, and Hualun Li. 2015. A Stochastic Model of Active Cyber Defense Dynamics. *Internet Mathematics* 11, 1 (2015), 23–61.
Shouhuai Xu, Wenlian Lu, and Li Xu. 2012a. Push- and Pull-based Epidemic Spreading in Arbitrary Networks: Thresholds and Deeper Insights. *ACM Transactions on Autonomous and Adaptive Systems (ACM TAAS)* 7, 3 (2012), 32:1–32:26.
Shouhuai Xu, Wenlian Lu, Li Xu, and Zhenxin Zhan. 2014. Adaptive Epidemic Dynamics in Networks: Thresholds and Control. *ACM Transactions on Autonomous and Adaptive Systems (ACM TAAS)* 8, 4 (2014), 19.
Shouhuai Xu, Wenlian Lu, and Zhenxin Zhan. 2012b. A Stochastic Model of Multivirus Dynamics. *IEEE Transactions on Dependable and Secure Computing* 9, 1 (2012), 30–45.
Shouhuai Xu, Haifeng Qian, Fengying Wang, Zhenxin Zhan, Elisa Bertino, and Ravi S. Sandhu. 2010. Trustworthy Information: Concepts and Mechanisms. In *Proceedings of 11th International Conference on Web-Age Information Management (WAIM'10)*. 398–404.
Shouhuai Xu, Ravi S. Sandhu, and Elisa Bertino. 2009. TIUPAM: A Framework for Trustworthiness-Centric Information Sharing. In *Proceedings of International Conference on Trust Management (IFIPTM'09)*. 164–175.
Shouhuai Xu and Moti Yung. 2009. Expecting the Unexpected: Towards Robust Credential Infrastructure. In *Financial Cryptography and Data Security, 13th International Conference, FC 2009, Accra Beach, Barbados, February 23-26, 2009. Revised Selected Papers*. 201–221.
Y. Yusuf, M. Sarhadi, and A. Gunasekaran. 1999. Agile manufacturing: the drivers, concepts and attributes. *International Journal of Production Economics* 62 (1999).
B. P. Zeigler. 1990. High autonomy systems: concepts and models. In *Proceedings of AI, Simulation and Planning in High Autonomy Systems*. 2–7.
Zhenxin Zhan, Maochao Xu, and Shouhuai Xu. 2013. Characterizing Honeypot-Captured Cyber Attacks: Statistical Framework and Case Study. *IEEE Transactions on Information Forensics and Security* 8, 11 (2013), 1775–1789.
Zhenxin Zhan, Maochao Xu, and Shouhuai Xu. 2014. A Characterization of Cybersecurity Posture from Network Telescope Data. In *Proc. of the 6th International Conference on Trustworthy Systems (InTrust'14)*. 105–126.
Zhenxin Zhan, Maochao Xu, and Shouhuai Xu. 2015. Predicting Cyber Attack Rates With Extreme Values. *IEEE Transactions on Information Forensics and Security* 10, 8 (2015), 1666–1677.
Ren Zheng, Wenlian Lu, and Shouhuai Xu. 2015. Active Cyber Defense Dynamics Exhibiting Rich Phenomena. In *Proc. 2015 Symposium and Bootcamp on the Science of Security (HotSoS'15)*. 2:1–2:12.
Ren Zheng, Wenlian Lu, and Shouhuai Xu. 2016. Preventive and Reactive Cyber Defense Dynamics Are Globally Stable. In *Manuscript*.
A. Zolli and A. Healy. 2012. *Resilience: Why Thins Bounce Back*. Free Press.