Towards an Organizationally-Relevant Quantification of Cyber Resilience

Thomas Llansó
Johns Hopkins University
Applied Physics Laboratory
thomas.llanso@jhuapl.edu

Martha McNeil
Johns Hopkins University
Applied Physics Laboratory
martha.mcneil@jhuapl.edu

Abstract

Given the difficulty of fully securing complex cyber systems, there is growing interest in making cyber systems resilient to the cyber threat. However, quantifying the resilience of a system in an organizationally-relevant manner remains a challenge. This paper describes initial research into a novel metric for quantifying the resilience of a system to cyber threats called the Resilience Index (RI). We calculate the RI via an effects-based discrete event stochastic simulation that runs a large number of trials over a designated mission timeline. During the trials, adverse cyber events (ACEs) occur against cyber assets in a target system. We consider a trial a failure if an ACE causes the performance of any of the target system's mission essential functions (MEFs) to fall below its assigned threshold level. Once all trials have completed, the simulator computes the ratio of successful trials to the total number of trials, yielding RI. The linkage of ACEs to MEFs provides the organizational tie.

1. Introduction

There is increasing recognition that cyber systems can likely never be made fully secure [1]. A host of root causes contribute to this situation, including high system complexity, interconnectedness, and use of low-assurance components. The desire for missions and business functions that depend on such systems to succeed despite imperfect security gives rise to the idea of **cyber resilience**. For this paper, our working definition for cyber resilience is the ability of a cyber system to support organizational objectives by providing an acceptable level of performance for its mission essential functions (MEFs) in spite of adverse cyber events (ACEs). By MEFs, we mean that subset of a cyber system's use cases [2] that most directly support organizational functions and mission objectives. By ACEs, we refer to breaches of data integrity, confidentiality, or availability that could occur in a number of ways, such as a malicious cyber attack or physical attack on cyber assets, component failure, operator error, software or hardware bugs, and acts of God.

The remainder of this paper is organized as follows. First, we discuss related work and identify gaps, including a resilience quantification gap. Next, we describe the paper's contribution relative to the gaps, the Resilience Index (RI) metric and a simulator that computes RI for target systems, and we present an example run of an instantiation of the simulator. Finally, we discuss limitations and future work areas.

2. Related Work

2.1. Definitions

In addition to our working definition, the term cyber resilience has many other definitions. For example, NIST 800-160 [3] defines resilience as: "The ability to anticipate, withstand, recover from, and adapt to adverse conditions, stresses, attacks, or compromises on systems that use or are enabled by cyber resources regardless of the source." CNSSI 4009 [4] defines the term as, "The ability to prepare for and adapt to changing conditions and withstand and recover rapidly from disruptions. Resilience includes the ability to withstand and recover from deliberate attacks, accidents, or naturally occurring threats or incidents." Björck, et al. [5] define the term more succinctly as: "The ability to continuously deliver the intended outcome despite adverse cyber events." Our working definition is similar to these definitions; however, we specifically highlight MEFs, as MEFs are our connection from ACEs in a cyber system to organization/mission relevance of those ACEs.

2.2. Frameworks and Mechanisms.

A number of resilience-related cyber frameworks have emerged in recent years. For example, volume 2 of NIST Special Publication 800-160, "Systems Security Engineering Cyber Resiliency Considerations for the Engineering of Trustworthy Secure Systems" [3] lays out a set of resilience goals (Anticipate, Withstand, Recover, Adapt), more specific objectives (e.g., Prepare, Continue, Constrain), and techniques and approaches (e.g., Deception, Diversity, Redundancy).

URI: https://hdl.handle.net/10125/71470
978-0-9981331-4-0

NIST's "Framework for Improving Critical Infrastructure Cybersecurity" [6] introduces the notions of "Identify, Protect, Detect, Respond, Recover" as a means to categorize resilience mechanisms. The Carnegie Mellon University Software Engineering Institute created the CERT Resilience Management Model [7], which states "By improving operational resilience processes …, an organization can use the model to improve and sustain the resilience of mission-critical assets and services." Linkov, et al. [8] describe a cyber resilience matrix that includes a number of resilience-related practices. The US Department of Defense Cybersecurity Survivability Endorsement [9] defines a set of cyber survivability attributes arranged into Prevent, Mitigate, and Recover pillars. While such frameworks are helpful in structuring our thinking about resilience, they do not propose analytical techniques for measuring overall resilience, nor do they offer analytics that recommend the best combination of resilience mechanisms. In addition to organizing frameworks, other researchers focus on specific resilience mechanisms (e.g. [10]–[12]) or on resilience in narrower domains, such as cyber-physical control systems [13].

2.3. Quantification

While quantification of resilience in other fields (e.g., ecology [14], mechanical engineering [15]) is more mature, the study of cyber resilience quantification is in a nascent state. Linkov and Kott [16] hint at resilience quantification as follows, "Assuming two equally performing systems A and B subjected to an impact …that left both systems with an equal performance degradation, the resiliency of system A is greater if after a given period T it recovers to a higher level of performance than that of system B." For these authors, time to recover is the critical resilience quantity. The report "Partnering for Cyber Resilience" [17] suggests that use of Monte Carlo modeling could provide a useful basis for quantifying resilience. In fact, our inspiration for RI came both from this report and the analogy of retirement calculators that take such an approach for estimating the success rate of a person's retirement plans (e.g., Vanguard [18] and retirementsimulation.com [19]). Many variables (e.g., inflation rate, market return, taxation rate, funding of Social Security, sequence of returns, longevity), contribute to the uncertainty of whether a person will have sufficient funds throughout retirement. The Monte Carlo approach treats these quantities as random variables with associated distributions. Through a run of, for example, 10,000 simulation trials, such calculators randomly sample the distributions of these variables in a time-based simulation to compute an overall success rate score.

2.4. Risk and Resilience

Linkov et al. [8] recognize the relationship between cyber resilience and risk, "resilient systems should utilize generalizable concepts distinct from but complementary to risk assessment." In the case of malicious attack, we hypothesize a link between risk and resilience in the sense that high risk portions of a system as measured by their mission criticality and unmitigated exposure to anticipated threats should receive attention when considering cyber resilience mechanisms, as these are likely to be the most attractive to attackers.

There are many cyber risk frameworks and approaches in use today, such as NIST 800-30 [20], OCTAVE [21], and INFOSEC Institute's "Quantitative Risk Analysis" method [22]. The RI simulator discussed in this paper leverages a risk approach called BluGen [23] in order to provide risk quantification. In BluGen, the risk, r(a), of a cyber asset, a, is a function of the asset's exposure, e, and criticality, c, as shown in equation (1). Briefly, asset exposure is the ratio of the number of unmitigated threat capabilities mapped to an asset of a given type to the total number of applicable threat capabilities. Asset criticality is a measure of an asset's importance to the MEFs it supports.

$$r(a) = 1.0 - \frac{\sqrt{(e(a) - 1.0)^2 - (c(a) - 1.0)^2}}{\sqrt{2}} \quad (1)$$

The numerator of the second term in Equation (1) is the distance formula from an asset plotted in an x-y plane by its exposure and criticality scores to the point of highest risk, (1.0, 1.0), such as in a BluGen risk scatterplot. We divide by $\sqrt{2}$ to scale to the range 0.0 to 1.0, and we subtract the resulting quantity from 1.0 so that risk rises as we approach (1.0, 1.0). BluGen provides "credit" for those mitigations added to the system description that apply to previously unmitigated threat capabilities that affect assets in the system. The effect is to lower the exposure score for relevant assets, because the numerator of the ratio mentioned above is reduced as relevant mitigations are added.

3. Research Gaps and Contribution

Despite the work in cyber resilience as summarized in the related work section, several research gaps remain: (1) a quantitative measure of overall resilience of a cyber system as it relates to the MEFs that a system provides is lacking; (2) the optimal level of coordination between mission-level resilience ("can we accomplish mission/business functions") and cyber-level resilience ("can we keep the system up") is unexplored; (3) the most effective balance of automated vs. manual responses to ACEs is unknown; and (4) objective guidance on how to combine resilience mechanisms into a resulting architecture that provides a given measure of resilience is deficient.

The contribution of this paper is an approach called the "Resilience Index" (RI) for estimating the resilience of a cyber-intensive system to ACEs. The primary focus of RI is the first gap listed above. The organizational relevance is RI's tie to the MEFs of a target system, which in turn support higher level organizational objectives (Figure 1).

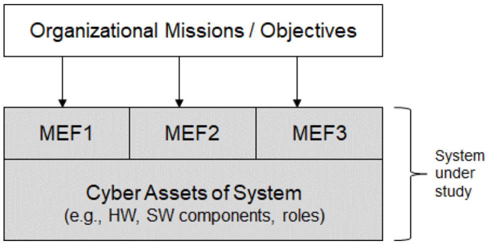

Organizational Missions / Objectives
MEF1
MEF2
MEF3
Cyber Assets of System
(e.g., HW, SW components, roles)
System under study

Figure 1. Organizational use of a cyber system

An example of Figure 1 is a satellite ground system shown in Figure 2. Section 6 below uses the example.

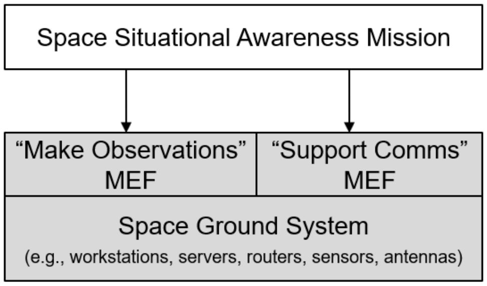

Space Situational Awareness Mission
"Make Observations" MEF
"Support Comms" MEF
Space Ground System
(e.g., workstations, servers, routers, sensors, antennas)

Figure 2. Space situational awareness example

Here, the higher-level mission is "space situational awareness (SSA)" with supporting ground system MEFs to (1) make observations and (2) facilitate SSA communications among ground parties. Each MEF, in turn, has an acceptable range of performance determined from organizational use of the system. (We note that some MEFs may have multiple associated metrics, or measures of effectiveness, but for simplicity, this paper assumes one metric per MEF.) MEF values are often expressed by threshold (minimally acceptable) and an objective (desired) values. Table 1 shows sample values for the SSA Mission MEFs (opm = observations per minute, mbps = megabits per second).

Table 1: Acceptable MEF value ranges

	MEF Values	
MEF	**Threshold**	**Objective**
Make Observations	10 opm	15 opm
Support Comms	25 mbps	75 mbps

Figure 3 illustrates how ACEs indirectly impact MEFs by affecting the data processed by system assets that ultimately affect MEF performance. By asset we mean a software or hardware component, or a system role played by a person. In the example, Asset 1 processes Data 1 in support of MEF2. If an ACE disrupts the integrity of Data 1 on Asset 1, MEF2 will be impacted.

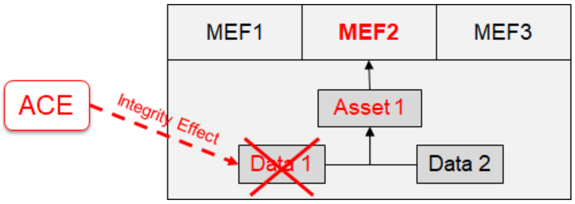


Figure 3. ACE-to-MEF relationship

Figure 4 illustrates the essence of cyber resilience. A given MEF can be impacted by one or more ACEs over a period of time. If the system is sufficiently resilient, MEF performance can be sustained to an acceptable level despite the ACE. The figure shows a system operating near its objective performance that encounters an ACE. Performance may degrade below the threshold for a time. The system or its operators may mount an active response to the detected ACE in order to reconstitute the system and recover the performance of the impacted MEF to an acceptable level of performance in a mission-relevant timeframe.

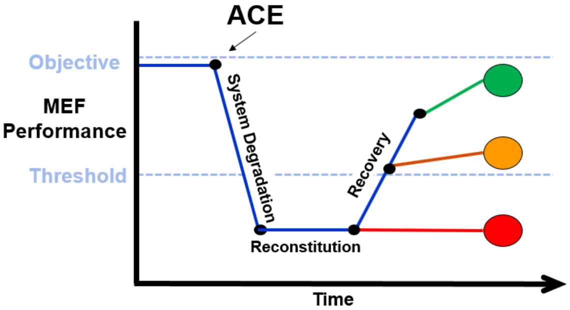


Figure 4. A cyber resilience illustration

This brings us to the meaning of the RI value, which, as discussed below, is the percentage of simulation trials in which none of the target system's MEF performance values dropped below their threshold values in spite of ACEs occurring during the trials.

4. Approach

We structured our creation and examination of RI in the context of Design Science Research (DSR) [24]. In addition to the RI metric itself, other DSR artifacts are the RI model and method, as summarized in the high-level architecture shown in Figure 5, and an initial instantiation of the model and method.

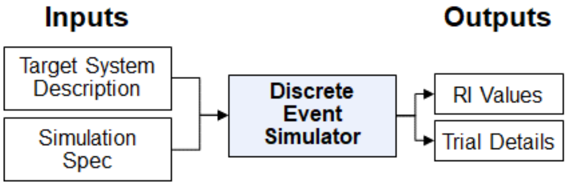


Figure 5. RI simulator inputs and outputs

We came to our overall approach by noting that there is considerable uncertainty with respect to the number, type, timing, and targeting of ACEs. Given the uncertainty, we constructed an effects-based discrete event stochastic simulation that treats such quantities as random variables. Unlike analogous approaches, such as retirement calculators, where decades of historical data are available to inform the random variable distributions, certain ACE-related variable distributions are not as readily available. We nonetheless designed our approach to accommodate such data as it becomes available. Meanwhile, in order to make initial progress, we employ uniform distributions for two variables. In other cases, we seek user input to supply values in "what if" scenarios.

Figure 6 provides an overview of a given run of the simulation. The idea is to execute n simulations or trials (Tr_1, Tr_2, …, Tr_n) during an overall run, where n is a configurable value large enough (e.g., 1,000) to provide reasonable coverage of value ranges of relevant variables. Each trial is in the context of an overall timeline, measured in discrete increments over an interval [*1, m*], where the user enters the end time, m.

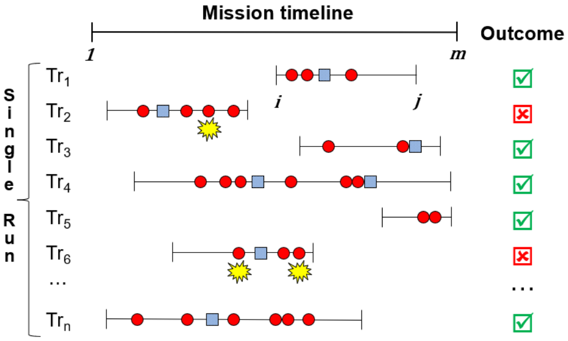


Figure 6. Simulation overview

The cyber system under study is considered to be operational and supporting higher level mission / business function via its MEFs during this time interval. The time units can be whatever is convenient to the organizational / mission context (e.g., seconds, minutes, hours, days). Each trial takes place over a sub interval, [*i, j*], of the total timeline, [*1, m*], with $1 \leq i \leq j \leq m$. During a trial, a number of ACEs take place, represented by red dots in Figure 6. The blue squares represent responses to ACEs.

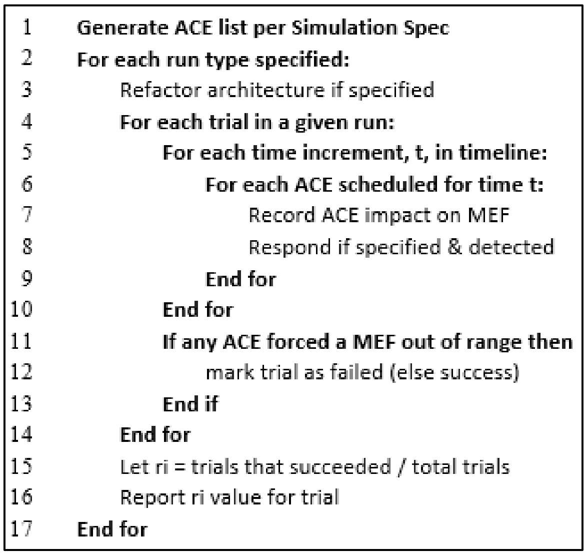
As described below, we model an ACE with respect to the effect it has on a particular asset in the target system and data processed by the asset. By effects-based, we mean that we focus on the effect the ACEs have rather than on the multitude of ways that any given type of ACE could come about. For example, in the case of a malicious ACE, many potential attack vectors and paths to a target asset and data may exist, with numerous potential vulnerabilities to exploit along the way, and possibly many different exploitation techniques available for use. We abstract this complexity away by focusing on the effects.

ACEs may or may not cause a failure of an overall trial. We define a failed trial as one in which the impact of an ACE that takes place during the trial exceeds the maximum allowed impact value for the MEF affected by the ACE. The yellow starbursts in Figure 6 indicate ACEs whose effects exceed the maximum impact allowable for the associated MEF, causing MEF performance to drop below its corresponding threshold value for some defined period of time. The result is a failed trial. We consider all other trials as successes. We define the RI for a system as the ratio of successful trials to the total number of trials, per equation (2).

$$RI(s) = \frac{Number\ of\ successful\ trials(s)}{Total\ number\ of\ trials(s)} \quad (2)$$

4.1. Simulation Method

Figure 7 summarizes the main simulation method. After identifying key variables in the simulation, we describe the steps in the simulation method in the sections following the figure.



```
Generate ACE list per Simulation Spec
For each run type specified:
    Refactor architecture if specified
    For each trial in a given run:
        For each time increment, t, in timeline:
            For each ACE scheduled for time t:
                Record ACE impact on MEF
                Respond if specified & detected
            End for
        End for
        If any ACE forced a MEF out of range then
            mark trial as failed (else success)
        End if
    End for
    Let ri = trials that succeeded / total trials
    Report ri value for trial
End for
```

Figure 7. Simulation method (algorithm)

The variables shown in Table 2 govern the simulation. We treat some as random variables, while others that currently depend on user entry (e.g., ACE detection probabilities) may be converted to random variables in the future as suitable distributions become available based on accumulated empirical data. The sections below discuss the variables in more detail.

The simulation method begins by generating a list of ACEs to execute during the trials of the simulation runs (line 1 in Figure 7). The simulator formulates ACEs based on a set of "criticality" tuples that are input to the RI simulator as part of the target system description (Figure 8). These criticality tuples are similar to those described in [23]. We limit ACEs to the possibilities defined in the list of criticality tuples because they represent the mapping of data to assets and assets to MEFs. They also define MEF impacts should an ACE occur in the context of the criticality tuple.

Table 2. Key simulation variables

Variable	Value Source
Environment to analyze	Input file or generated
Run types to execute	User entry
Number of trials per run	User entry
Number of ACEs per trial	User entry
ACE candidates for application during a run	Drawn from uniform distribution over input criticality set
Timing of ACE	Drawn from uniform distribution over criticality time range
ACE relative event probability by ACE type	User entry
ACE detection probability by C/I/A	User entry
Risk to Asset	See equation (1)

As Figure 8 indicates, a criticality is a 7-tuple entity of the form (MEF, Asset, Data, Begin Time, End Time, Effect, Impact Score). One can derive such data from experimentation, simulation [25], or, often, subject matter expert interviews.

The meaning of a tuple instance is that an adverse cyber event against a given data item processed by the given cyber asset that supports a given MEF at any time during over a given time interval will have an impact on the MEF performance indicated by the given score. Scores range from 0.0 (no effect) to 1.0 (MEF performance is fully compromised). The effect is either a breach of data confidentiality, integrity, or availability.

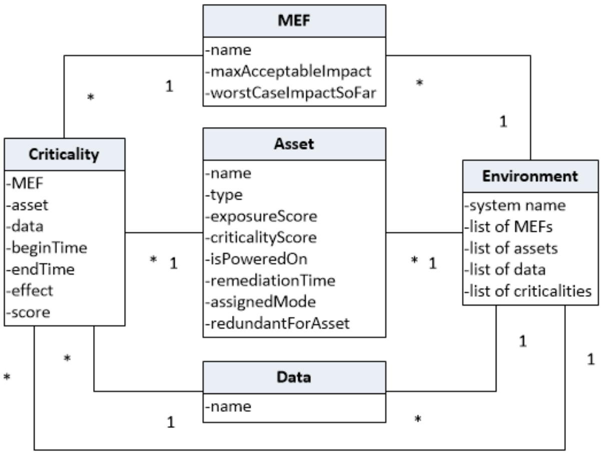


Figure 8. Describing a target system for analysis

Figure 9 defines an ACE, which is an assigned ACE type (described below), a selected criticality 7-tuple, and a scheduled time to occur.

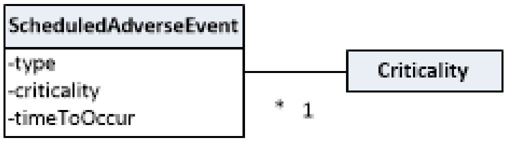


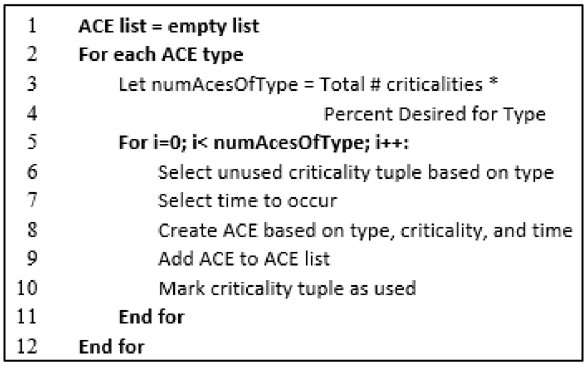
Figure 9. Scheduled ACE

ACE types are defined in Table 3. The RI simulator selects and schedules ACEs based on user specifications, as described below. An overview of the algorithm to generate the list of ACEs per the simulation specification appears in Figure 10. Line 6 of Figure 10 selects an unused criticality tuple based on the ACE type. For a malicious attack ACE type, the simulator selects the criticality tuple with the highest computed risk, as defined in equation (1). As mentioned earlier, this data is provided as input to the RI simulator and can be produced by risk methods such as BluGen [23].

Table 3. ACE types

ACE Type	Example Influencing Factors
Malicious Attack	Motivation level, stealth concern, asset risk (risk to asset owners)
Operator Error	Operator training, experience, assignments in the system
Asset Failure	Asset time in service, mean time between failure (MTBF)
Acts of God	Asset geographic location, time of year, climate change
Bugs in SW/HW	Organizational CMMI level [26]

For ACE types other than malicious attack, the current simulator implementation makes a random selection from among the unused criticalities based on a uniform distribution. A future implementation could use other approaches. For example, for the asset failure ACE type, one might select an unused criticality tuple whose asset has the highest mean time between failure (MTBF) based on its asset type.



```
ACE list = empty list
For each ACE type
    Let numAcesOfType = Total # criticalities *
                                  Percent Desired for Type
    For i=0; i< numAcesOfType; i++:
        Select unused criticality tuple based on type
        Select time to occur
        Create ACE based on type, criticality, and time
        Add ACE to ACE list
        Mark criticality tuple as used
    End for
End for
```

Figure 10. Algorithm to Generate Initial ACE List

4.2. Sensitivity Analysis

One purpose of the RI simulator is to allow for sensitivity analysis of the computed RI values under various transformations of the system under analysis and/or active responses taken during a simulation run. The loop setting up the runs appears in line 2 of Figure 7. To support sensitivity analysis, the simulation applies the ACEs to the target system in a series of runs or passes specified by the user. The runs facilitate sensitivity analysis with respect to how changes in initial assumptions affect the overall RI value computed for each run. The simulator is built to support different types of runs, as shown in Table 4.

Table 4. Available Run Types

Run Type	Description
Passive	A run where no responses to ACEs are mounted
Actively Respond	A run where active response actions are mounted for detected ACEs (not all ACEs are detected)
Refactor	A run where the architecture of the system is refactored in some way
Change Risk	A run where asset risks are modified by some percentage (either up or down) to see the resilience impact for malicious ACEs

Below, we discuss two run types from Table 4**,** active responses to ACEs and architectural refactoring.

4.2.1. Active Responses to ACEs

Per line 8 in Figure 7, a run may involve active response to ACEs (the blue squares in Figure 6). There is a range of potential response actions to ACEs. In the RI simulator, we chose to initially focus on a key mechanism discussed in our 2016 paper on resilience in a space ground system [27]. The mechanism involves having multiple alternative modes of operation. The primary mode is the default mode and provides the best mission performance, but is also the most complex and interconnected. If an ACE within the components that make up the primary mode is detected and deemed serious enough, the system operator may opt to bring down the primary mode and bring up a secondary mode of operation that is higher assurance but does not offer the same level of performance. This multi-mode arrangement provides time to remediate assets tied to the original mode before bringing the original mode back on-line while still sustaining an acceptable level of mission performance during remediation. Depending on the system, additional modes may be justified.

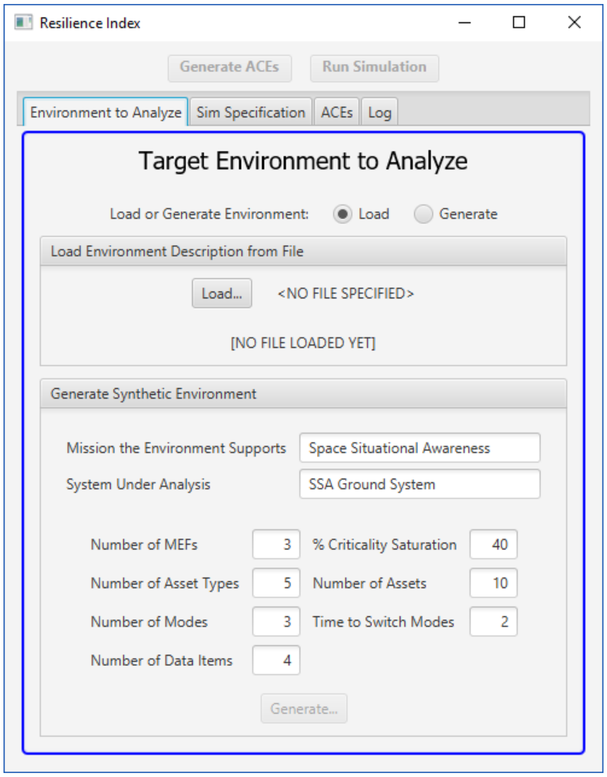
4.2.2. Architectural Refactoring

The simulator has a limited ability to automatically refactor the architecture of a target environment to assess its impact on the RI score. The simulator is currently built to support the use of redundancy and deception. For the redundancy approach, the simulator can introduce redundant assets into the architecture for assets that exceed a user-specified criticality threshold value for criticality tuples that have an availability effect. The user specifies the number of new assets to introduce into the architecture and the amount of criticality reduction the simulator should apply to the original and newly introduced assets. For example, suppose the user does the following: opts for redundancy refactoring, sets the associated criticality threshold to 0.8, sets the number of new assets to introduce at 2, and sets the criticality reduction value to 0.2 (a 20% reduction). Then suppose the simulator encounters an asset, a_1, with a criticality value of 0.9 for availability. The simulator would introduce two new assets, a_2 and a_3, into the architecture and criticality tuple list, and set the criticality of a_1, a_2, and a_3 to 0.7 for the availability effect.

For deception, which is only partially implemented in the instantiation, the idea is to introduce decoys assets into the architecture for assets that have a criticality that exceeds a certain threshold value. Use of decoy assets can aid in detecting adversary presence. If the decoy assets can switch back and forth from being decoy assets to redundant assets under a specified (and unpredictable to the adversary) schedule, then they may contribute to a criticality reduction for the original assets.

5. Model and Method Instantiation

The current instantiation of the RI model and simulator method is a Java program with a user interface that appears in figures below. The simulator uses a tabbed interface. The first tab (Figure 11) allows the user to specify the target environment to analyze.



Target Environment to Analyze

Load or Generate Environment: ● Load ○ Generate

Load Environment Description from File

<NO FILE SPECIFIED>

[NO FILE LOADED YET]

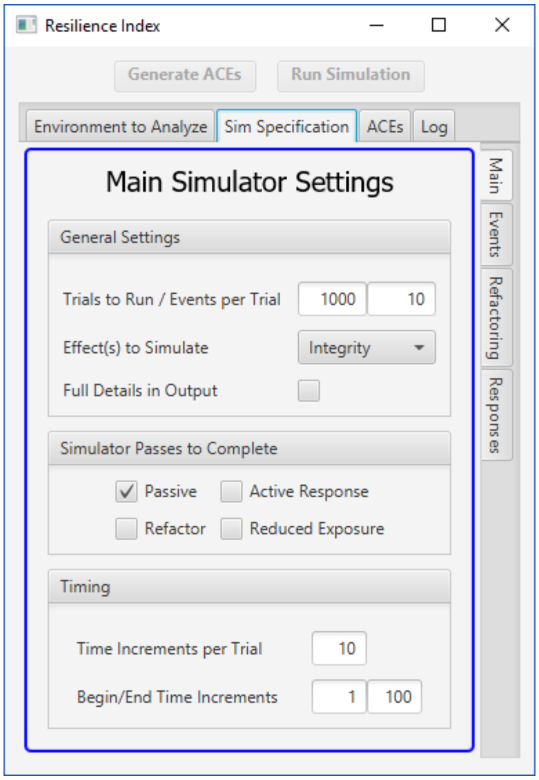
Generate Synthetic Environment

Mission the Environment Supports	Space Situational Awareness		
System Under Analysis	SSA Ground System		
Number of MEFs	3	% Criticality Saturation	40
Number of Asset Types	5	Number of Assets	10
Number of Modes	3	Time to Switch Modes	2
Number of Data Items	4		

Figure 11. Environment tab

The user can load the target environment to be analyzed from a file-based system description formatted as a structured comma-separated value (CSV) file, or generate a synthetic environment based on a set of parameters for experimentation purposes.

The next tab, Sim Specification (Figure 12), allows the user to specify how the simulator is to run. The tab contains four sub-tabs down the right side: Main, Events, Refactoring, and Responses. In the main tab, the user sets the total number of trials to run, the number of ACEs per trial, the effects to simulate (Confidentiality, Integrity, Availability, or some combination of these effects) during the ACEs, which passes the simulator is to make to allow comparison of different RI values for the same target system under different system conditions. Finally, the user can specify the overall length of the trials, and which subinterval the simulation should choose from within for the span of simulated ACEs.



Main Simulator Settings

General Settings

Trials to Run / Events per Trial	1000	10
Effect(s) to Simulate	Integrity ▾	
Full Details in Output	□	

Simulator Passes to Complete

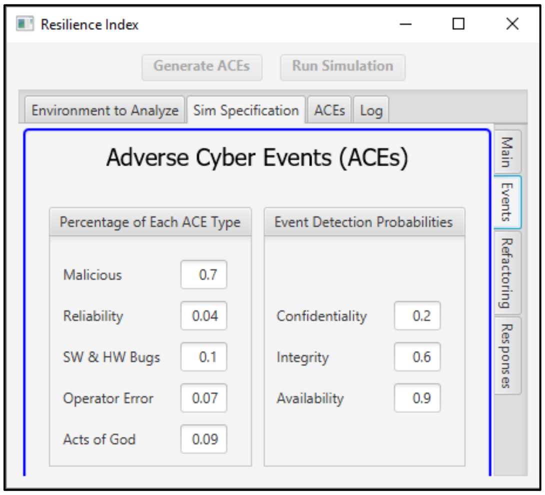
☑ Passive □ Active Response

□ Refactor □ Reduced Exposure

Timing

Time Increments per Trial	10	
Begin/End Time Increments	1	100

Figure 12. Sim. specification—main tab



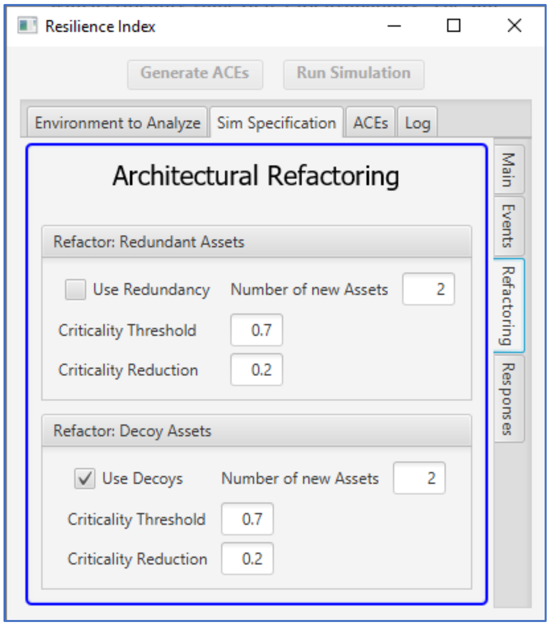
Adverse Cyber Events (ACEs)

Percentage of Each ACE Type		Event Detection Probabilities	
Malicious	0.7		
Reliability	0.04	Confidentiality	0.2
SW & HW Bugs	0.1	Integrity	0.6
Operator Error	0.07	Availability	0.9
Acts of God	0.09		

Figure 13. Events tab

Figure 13 shows the Events tab, which allows the user to specify relative probabilities of the five ACE types and event detection probabilities for breaches of confidentiality, integrity, and availability.

The refactoring tab (Figure 14) provides an initial exploration of refactoring possibilities centered around the introduction of redundant assets and decoy assets into the architecture, as discussed earlier.



Architectural Refactoring

Refactor: Redundant Assets

☐ Use Redundancy Number of new Assets 2

Criticality Threshold 0.7

Criticality Reduction 0.2

Refactor: Decoy Assets

☑ Use Decoys Number of new Assets 2

Criticality Threshold 0.7

Criticality Reduction 0.2

Figure 14. Refactoring tab

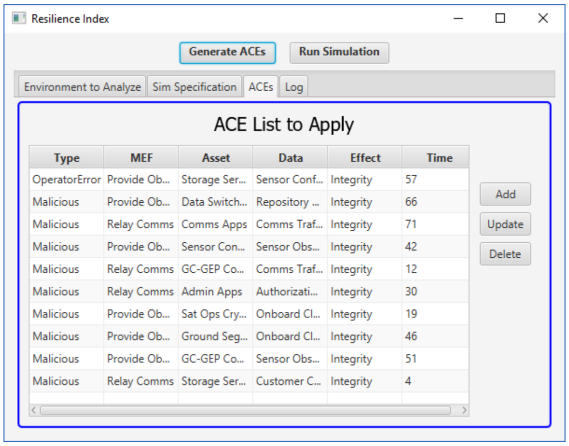
The Responses tab (not shown) allows the user to indicate whether to use mode switching as an ACE response mechanism, as discussed earlier.

6. Example

We tested the initial simulator instantiation against a ground system that controls a geosynchronous satellite and its optical sensing payload (this example was briefly referenced in Figure 2). For sensitivity reasons, the system is an exemplar only, but experienced ground segment engineers designed the exemplar system to be realistic, similar to actual deployed ground systems. The higher-level mission is space situational awareness. In support of this mission, the system has two primary MEFs, as shown in Figure 2, and has 39 assets that support the MEFs. The assets, in turn, collectively process 26 data types, and there are 1,010 distinct criticality tuples that specify the mission criticality of different combinations of MEF, asset, data, times, and compromise effects.

We ran 1,000 trials with 10 ACEs per trial over a total timeline of 100 time increments. We considered a single time increment to be one day. For simulator runs, we chose only the passive case, where ACEs occur, but no active response is mounted. The ACEs generated appear in Figure 15. As shown, 9 out of 10 ACEs are malicious attacks, with one reliability failure on the storage server asset. This particular mix of ACEs resulted from ACE type probabilities that we specified of Malicious=0.70, reliability=0.04, bugs=0.10, operator error=0.07, and acts of God=0.09.

While we could have custom-edited the starting ACE list, we opted to accept the generated list.

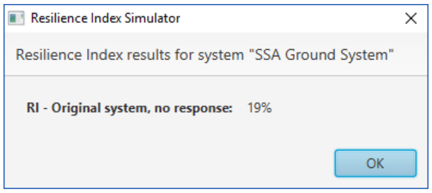


ACE List to Apply

Type	MEF	Asset	Data	Effect	Time
OperatorError	Provide Ob...	Storage Ser...	Sensor Conf...	Integrity	57
Malicious	Provide Ob...	Data Switch...	Repository ...	Integrity	66
Malicious	Relay Comms	Comms Apps	Comms Traf...	Integrity	71
Malicious	Provide Ob...	Sensor Con...	Sensor Obs...	Integrity	42
Malicious	Relay Comms	GC-GEP Co...	Comms Traf...	Integrity	12
Malicious	Relay Comms	Admin Apps	Authorizati...	Integrity	30
Malicious	Provide Ob...	Sat Ops Cry...	Onboard Cl...	Integrity	19
Malicious	Provide Ob...	Ground Seg...	Onboard Cl...	Integrity	46
Malicious	Provide Ob...	GC-GEP Co...	Sensor Obs...	Integrity	51
Malicious	Relay Comms	Storage Ser...	Customer C...	Integrity	4

Figure 15. Generated ACE list

The result of running the simulator against this example appears in Figure 16. As shown, the Resilience Index of the exemplar system is only 19%.



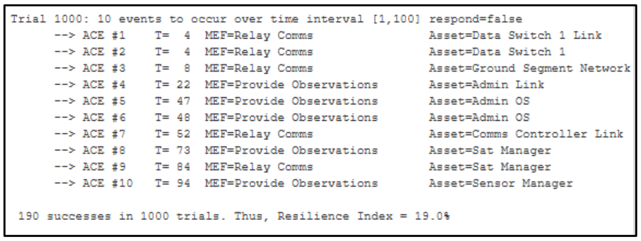
Resilience Index results for system "SSA Ground System"

RI - Original system, no response: 19%

Figure 16. RI Simulator results summary

The low RI value represents an initial case before architectural refactoring, improved mitigations (resulting in lower exposure scores), and active response are done. Such activities could raise the RI value, but not all of these features are part of the instantiation yet.

A portion of the simulator log showing the final trial appears in Figure 17. In the trial, three ACEs (#'s 4, 5, and 6) had impact scores for the "Provide Observations" MEF that exceeded the allowable threshold of 0.8, resulting in the failure of that particular trial. The simulator took 5 minutes and 49 seconds to run on a Windows 10 desktop computer with an Intel I7 processor and 16GB of memory. Note that at this stage we have not attempted any kind of performance optimizations on the simulator instantiation.



```
Trial 1000: 10 events to occur over time interval [1,100] respond=false
      --> ACE #1    T=  4  MEF=Relay Comms                Asset=Data Switch 1 Link
      --> ACE #2    T=  4  MEF=Relay Comms                Asset=Data Switch 1
      --> ACE #3    T=  8  MEF=Relay Comms                Asset=Ground Segment Network
      --> ACE #4    T= 22  MEF=Provide Observations       Asset=Admin Link
      --> ACE #5    T= 47  MEF=Provide Observations       Asset=Admin OS
      --> ACE #6    T= 48  MEF=Provide Observations       Asset=Admin OS
      --> ACE #7    T= 52  MEF=Relay Comms                Asset=Comms Controller Link
      --> ACE #8    T= 73  MEF=Provide Observations       Asset=Sat Manager
      --> ACE #9    T= 84  MEF=Relay Comms                Asset=Sat Manager
      --> ACE #10   T= 94  MEF=Provide Observations       Asset=Sensor Manager

 190 successes in 1000 trials. Thus, Resilience Index = 19.0%
```

Figure 17. Final section of simulator log

7. Discussion and Evaluation

Revisiting the RI simulator through the lens of Design Science Research (DSR), and the Peffers et al. DSR model [24] in particular, we have (1) motivated the need for cyber resilience given the challenges of making complex, interconnected cyber systems fully secure, (2) defined a concept for resilience quantification that manifests as an effects-based discrete event stochastic simulation, (3) demonstrated an instantiation of the simulation against a realistic cyber system, (4) conducted a preliminary assessment of the model, method, and instantiation artifacts, and (5) communicated initial results through this paper.

There are a number of limitations related to this initial work: (1) We speculate that simulated malicious ACE results and attacker behavior converge over the long run, but more data is required to evaluate this idea; (2) the simulator uses a simplified model of a target cyber system, not taking, for example, component connectivity into account; (3) The simulator does not currently offer a full array of automated refactoring and ACE response options, including game-theoretic interactions, that would allow the user to more fully explore the resilience "tradespace"; (4) The simulator depends on the user to specify certain values for certain variables used by the simulator, such as ACE type and detection probabilities. While specifying such probabilities allows for values that may vary by system context, providing empirically-based initial default values could be of value; (5) Validation exploration of the RI simulator has not yet been undertaken.

8. Conclusions and Future Work

Through its connection to MEFs, the RI simulator described in this paper provides, we argue, an organizationally relevant quantification of cyber resilience, addressing the quantification gap mentioned earlier. The simulator represents an initial foundation for future work in resilience exploration.

A number of future work possibilities exist: (1) improve the modularity of the design to allow new refactoring and response options to be plugged into the simulator and automatically explored if specified by the user; (2) incorporate time-to-recover to model the recovery of assets taken offline; (3) automatically explore combinations of refactoring and response actions to identify the highest RI scoring possibilities; (4) incorporate other tradespace priorities into the sensitivity analysis for various options, such as costing constraints; (5) calibrate simulator defaults with the results of empirical experiments (e.g., the degree to which using quantified risk estimates to prioritize malicious ACE targeting is reflected in attacker behavior); (6) enhance metadata about target architectures with data that informs ACE default probabilities (e.g., asset type-specific mean time between failures for reliability-related ACEs, training data to inform operator error ACEs.); (7) carry out evaluations of artifact utility with an appropriate target audience, and (8) incorporate simulated human decision making in mounting certain responses; (9) as data becomes available, revisit the use of uniform distributions in the simulator; in the meantime, explore the potential use of other distributions and their impact; (10) consider other effects in the simulator, such as confidentiality, non-repudiation, and authentication choices; (11) give further thought to approaches for validating RI simulator results, such as by calibrating the RI simulator variables and the ACE list generation approach based on data gathered from long-term observational studies of cyber systems operating in a realistic threat environment; (12) give consideration for how to model the effects of simultaneous ACEs; (13) explore the idea of introducing into the specification a minimum downtime for an ACE before the effect of the ACE is registered (that is, rapid recovery from an ACE might negate the MEF impacts); (14) explore expanding the simulator to incorporate into resilience the idea of maintaining not just threshold level MEF performance but performance that sustainably approaches or meets objective MEF performance.

9. Acknowledgments

The authors wish to acknowledge and thank the early reviewers of this paper, including Dallas Pearson, Luanne Chamberlain, and Peter Dinsmore.

10. References

[1] P. Denning and D. Denning, "Cybersecurity Is Harder Than Building Bridges," *American Scientist*, vol. 104(3), pp. 154–157, 2016.

[2] G. Booch, J. Rumbaugh, and I. Jacobson, *The Unified Modeling Language User Guide (2nd Edition)*. Addison-Wesley, 2005.

[3] R. Ross, R.; Graubart, R.; Bodeau, D.; McQuaid, "Systems Security Engineering Cyber Resiliency Considerations for the Engineering of Trustworthy Secure Systems," 2018.

[4] Committee on National Security Systems, "CNSS 4009 - National Information Assurance (IA) Glossary," 2010. [Online]. Available: http://www.ncsc.gov/publications/policy/docs/CNSSI_4009.pdf.

[5] F. Björck, M. Henkel, J. Stirna, and J. Zdravkovic, "Cyber Resilience -- Fundamentals for a Definition," in *New Contributions in Information*

Systems and Technologies, 2015, pp. 311–316.

[6] NIST, “Framework for Improving Critical Infrastructure Cybersecurity, Version 1.1, Draft 2,” 2017.

[7] Software Engineering Institute, “CERT Resilience Management Model (CERT-RMM),” 2016.

[8] I. Linkov, D. A. Eisenberg, K. Plourde, T. P. Seager, J. Allen, and A. Kott, “Resilience metrics for cyber systems,” *Environ. Syst. Decis.*, vol. 33, no. 4, pp. 471–476, Dec. 2013.

[9] D. Davidson, “New DoD Approach on the Cyber Survivability of Weapon Systems,” 2017. [Online]. Available: https://ndiastorage.blob.core.usgovcloudapi.net/ndia/2017/systems/Wednesday/Track1/19870_Davidson.pdf.

[10] H. Okhravi, J. Haines, and K. Ingols, “Achieving Cyber Survivability in a Contested Environment Using a Cyber Moving Target,” *High Front. J. Sp. Cybersp. Prof.*, vol. 7, no. 3, pp. 9–13, 2011.

[11] Y. Yuan, Q. Zhu, F. Sun, Q. Wang, and T. Başar, “Resilient control of cyber-physical systems against Denial-of-Service attacks,” in *2013 6th International Symposium on Resilient Control Systems (ISRCS)*, 2013, pp. 54–59.

[12] Y. Hayel and Quanyan Zhu, “Resilient and secure network design for cyber attack-induced cascading link failures in critical infrastructures,” in *2015 49th Annual Conference on Information Sciences and Systems (CISS)*, 2015, pp. 1–3.

[13] C. Alcaraz, “Cloud-Assisted Dynamic Resilience for Cyber-Physical Control Systems,” *IEEE Wirel. Commun.*, vol. 25, no. 1, pp. 76–82, Feb. 2018.

[14] D. Angeler and C. Allen, “Quantifying Resilience,” *J. Appl. Ecol.*, 2016.

[15] N. Yodo and W. Pingfeng, “Engineering Resilience Quantification and System Design Implications,” *J. Mech. Des.*, 2016.

[16] I. Kott, Alexander; Linkov, Ed., *Cyber Resilience of Systems and Networks*. Springer International Publishing, 2018.

[17] World Economic Forum, “Partnering for Cyber Resilience Towards the Quantification of Cyber Threats,” 2015.

[18] Vanguard, “Retirement Nest Egg Calculator,” 2019. [Online]. Available: https://retirementplans.vanguard.com/VGApp/pe/pubeducation/calculators/RetirementNestEggCalc.jsf.

[19] Retirement Simulation, “Monte Carlo Retirement Calculator.” [Online]. Available: https://www.retirementsimulation.com.

[20] National Institute of Standards and Technology, “National Institute of Standards and Technology 800-30: Guide for Conducting Risk Assessments,” 2012.

[21] R. Caralli, J. Stevens, L. Young, and W. Wilson, “Introducing OCTAVE Allegro: Improving the Information Security Risk Assessment Process,” 2007. [Online]. Available: http://www.cert.org/octave.

[22] INFOSEC Institute, “Quantitative Risk Analysis,” 2013. [Online]. Available: http://resources.infosecinstitute.com/quantitative-risk-analysis/#gref.

[23] M. McNeil, T. Llanso, and D. Pearson, “Application of Capability-Based Cyber Risk Assessment Methodology to a Space System,” in *Hot Topics in the Science of Security Symposium*, 2018.

[24] K. Peffers, T. Tuunanen, M. A. Rothenberger, and S. Chatterjee, “A Design Science Research Methodology for Information Systems Research,” *J. Manag. Inf. Syst.*, vol. 24, no. 3, pp. 45–77, Dec. 2007.

[25] T. Llanso and E. Klatt, “CyMRisk: An approach for computing mission risk due to cyber attacks,” in *IEEE International Systems Conference. Ottawa*, 2014, pp. 1–7.

[26] M. Paulk, C. Weber, S. Garcia, M. B. Chrissis, and M. Bush, “Key Practices of the Capability Maturity ModelSM, Version 1.1,” 1993.

[27] T. Llanso and D. Pearson, “Achieving Space Mission Resilience to Cyber Attack: Architectural Implications,” in *American Institute of Aeronautics and Astronautics Space 2016*, 2016, p. 11.