ARL-TR-9672 ● APR 2023

DEVCOM ARMY RESEARCH LABORATORY

A Methodology for Quantitative Measurement of Cyber Resilience (QMOCR)

by Alexander Kott, Michael J Weisman,
Joachim Vandekerckhove, Jason E Ellis, Travis W Parker,
Brian J Murphy, and Sidney Smith

NOTICES

Disclaimers

The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

Citation of manufacturer's or trade names does not constitute an official endorsement or approval of the use thereof.

Destroy this report when it is no longer needed. Do not return it to the originator.

ARL-TR-9672 ● APR 2023

A Methodology for Quantitative Measurement of Cyber Resilience (QMOCR)

Alexander Kott, Michael J Weisman, Jason E Ellis, and Sidney Smith
DEVCOM Army Research Laboratory

Travis W Parker
ICF International

Joachim Vandekerckhove
University of California, Irvine

Brian J Murphy
Pennsylvania State University

REPORT DOCUMENTATION PAGE

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED	
		START DATE	END DATE
April 2023	Technical Report	January 2021	March 2023

4. TITLE AND SUBTITLE
A Methodology for Quantitative Measurement of Cyber Resilience (QMOCR)

5a. CONTRACT NUMBER	5b. GRANT NUMBER	5c. PROGRAM ELEMENT NUMBER
5d. PROJECT NUMBER	**5e. TASK NUMBER**	**5f. WORK UNIT NUMBER**

6. AUTHOR(S)
Alexander Kott, Michael J Weisman, Joachim Vandekerckhove, Jason E Ellis, Travis W Parker, Brian J Murphy, and Sidney Smith

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
DEVCOM Army Research Laboratory
ATTN: FCDD-RLD
Adelphi, MD 20783

8. PERFORMING ORGANIZATION REPORT NUMBER
ARL-TR-9672

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release: distribution unlimited.

13. SUPPLEMENTARY NOTES
ORCID IDs: Alexander Kott, 0000-0003-1147-9726; Michael J Weisman, 0000-0003-4918-5571; Sidney Smith, 0000-0003-1398-307X

14. ABSTRACT
This report describes a methodology for measuring—quantitatively and experimentally—the cyber resilience of a system when subjected to a cyber attack. We use the term Quantitative Measurement of Cyber Resilience (QMOCR) to refer to this methodology.
The methodology is an outcome of the eponymous research project performed by the US Army Combat Capabilities Development Command Army Research Laboratory during the period of April 2021 through March 2023 in collaboration with the Pennsylvania State University Applied Research Laboratory and the University of California, Irvine. Resilience continues to gain attention as a key property of cyber and cyber-physical systems, for the purposes of cyber defense. A key challenge in the field of cyber resilience is quantifying or measuring resilience. Developers and buyers of a system must be able to quantify the cyber resilience of the system they develop or purchase. We recommend application of the QMOCR methodology especially when an actual system, its prototype, or a working, executable model (physical, digital, or digital–physical) is available.

15. SUBJECT TERMS
Network, Cyber, and Computational Sciences; cyber resilience; cyber experimentation; resilience modeling; cyber-physical systems

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES
a. REPORT	b. ABSTRACT	C. THIS PAGE		
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UU	36

19a. NAME OF RESPONSIBLE PERSON	19b. PHONE NUMBER *(Include area code)*
Alexander Kott	(301) 394-1507

STANDARD FORM 298 (REV. 5/2020)
Prescribed by ANSI Std. Z39.18

Contents

List of Figures iv

Acknowledgments v

1. Introduction 1

2. Key Steps of the Methodology 2

3. Detailed Discussion 3

4. Conclusions and Recommendations 9

5. References 11

Appendix. Quantitative Measurement of Cyber Resilience: Modeling and Experimentation 12

List of Symbols, Abbreviations, and Acronyms 27

Distribution List 28

List of Figures

Fig. 1 Computing the relative functionality 7

Acknowledgments

The authors would like to thank the Office of the Under Secretary of Defense Research and Engineering for funding the Quantitative Measurement of Cyber Resilience project.

1. Introduction

This report describes a methodology for measuring—quantitatively and experimentally—the cyber resilience of a system when subjected to a cyber attack. We use the term Quantitative Measurement of Cyber Resilience (QMOCR) to refer to this methodology.

The methodology is an outcome of the eponymous research project performed by the US Army Combat Capabilities Development Command (DEVCOM) Army Research Laboratory (ARL) during the period of April 2021 through March 2023 in collaboration with the Pennsylvania State University Applied Research Laboratory and the University of California, Irvine.

Resilience continues to gain attention as a key property of cyber and cyber-physical systems, for the purposes of cyber defense. Although definitions vary, it is generally agreed that cyber resilience refers to the ability of a system to resist and recover from a cyber compromise that degrades the business task-relevant performance of the system (Kott and Linkov 2019; Smith 2023). Resilience should not be conflated with security or risk management (Linkov et al. 2018).

A key challenge in the field of cyber resilience is quantifying or measuring resilience. Indeed, no engineering discipline achieved significant maturity without being able to measure the properties of phenomena relevant to the discipline (Kott and Linkov 2021). Developers of a system must be able to quantify the cyber resilience of the system under development in order to know whether the features they introduce in the system improve its cyber resilience or make it worse. Similarly, buyers of the system need to know how to quantitatively specify and experimentally test the system's cyber resilience in order to determine whether the product meets their specifications.

Throughout our research and in this report, we use the term "measure" or "measurement" as opposed to the terms "metric" or "assessment." We do so even though the term metric is quite popular. Typically (although not uniformly) within the cyber resilience literature, metrics refer to qualitative assessments of a system (actually existing or its design) by subject-matter experts (SMEs) (Alexeev et al. 2017; Linkov et al. 2013; Beling et al. 2021).

We, however, take a different perspective: quantitative and not qualitative, experimental, using physical quantities to the extent possible, business task focused, theoretically and empirically grounded. As such, we chose to use the term "measure" and not "metrics."

Section 2 proceeds to summarize, in a concise fashion, the key steps of the QMOCR methodology. This is intended to provide the reader with the gist of the approach, for the purposes of quick initial familiarization. Then, Section 3 provides more detailed discussion of each step. Section 4 offers conclusions and recommendations. The Appendix defines key concepts used in this report and describes an example of an experimental test-bed and an experimental technique that can serve as a simple example of applying the QMOCR methodology. It also discusses the mathematical techniques used to process the experimental data within the QMOCR methodology.

2. Key Steps of the Methodology

Here we outline, briefly, the key steps that together constitute the QMOCR methodology. The following section provides more detailed discussion of each step.

1) Identify, document, and obtain a system under test (SUT).
2) Define a set of representative business tasks of the SUT.
3) Define an appropriate, representative set of cyber attacks.
4) Define aggregate business task-relevant performance function of the SUT.
5) Equip the SUT with instrumentation for data collection.
6) Develop tools that allow the testing team to execute (or emulate effects of) the cyber attacks in a repeatable fashion.
7) Execute a sample of business tasks nominally (no cyber attacks); measure business task-relevant performance parameters.
8) Repeat same business tasks while undergoing a (randomized) set of cyber attacks; measure business task-relevant performance parameters.
9) Compute relative functionality for each business task.
10) Compute the measure of resilience R, for each attack episode within the business task.
11) Compute effectiveness of malware M, and effectiveness of bonware B for each attack episode within the business task.
12) Review and document the meaning of the results.

3. Detailed Discussion

This section is organized as a collection of notes referred to in the previous section. The notes explain and illustrate the key steps of the QMOCR methodology.

The numbering of notes corresponds to the steps outlined in Section 2. For example, Notes 3a and 3b refer to Step 3, and so on.

Note 1. The SUT can be an actual system, at different levels of development ranging from an early prototype to a deployed system. Alternatively, it could be a working, executable model (physical, digital, or digital–physical). As an example, in one case study (see Appendix) we used a digital–physical model of a cargo truck. The engine control units (ECUs) and the controller area network (CAN) bus of the truck were physical (i.e., actual electronic components). All other functional elements of the truck were computer simulated, including such things as performance of the engine and cooling system, interaction of the truck with the physical environment (terrain), additional sensors and their behaviors, and so on. In this case study, we also had an actual cargo truck for final testing and analysis of our methodology. In another case study, the SUT was a developmental database system for which a working prototype was available.

Note 2. The variability of business tasks that a given SUT might perform is often very broad. It is impossible to test the SUT under every possible variation of its business tasks. The testing team, in coordination with relevant organizations, should select a small set of typical business tasks (e.g., no more than 3 to 5 significantly different business task types) and define variable parameters associated with each business task. For example, in one case study (see Appendix) for a cargo truck, in consultations with SMEs we selected one type of a business task: delivering a cargo to a customer location, over a mountainous terrain. For this business task type, we selected several variable parameters including profile of the terrain (multiple routes were pseudo-randomly chosen from a representative 3-D terrain model), and quality of the road pavement (paved, gravel, unimproved trail).

In another case study, the types of possible business tasks were very few, and in consultation with SMEs we have considered only one, most common business task type, with timing of the adversarial attack being the only parameter.

Note 3a. The diversity of types of cyber attacks, the variability within a given cyber attack type, and combinations of attacks are infinite. Only a limited set can be explored in any realistically feasible test. The test team should consult relevant organizations for the types of cyber attacks that are considered representative and likely to be experienced by the given SUT within the given set of business tasks (as defined in Step 2). Where appropriate, the testing team, in coordination with

relevant organizations, should define variable parameters associated with each attack type within the selected set. For example, in a case study involving a cargo truck, we selected the following types of attacks: message flooding/signal takeover and ECU firmware alteration. Some of these attacks could be parameterized, such as time duration of attack and variable value assignment of a signal after successful takeover.

Note 3b. In some cases, it is more productive to define not the specific attacks but rather their effects on the SUT. This approach offers a major advantage: instead of testing a multiplicity of cyber attacks, the test can focus on the potential effects produced by classes of attacks on the same manifestation of resistance and recovery exhibited (or not) by the SUT. For example, in the case study of a digital–physical model of a cargo truck, we determined that an extremely broad range of attacks (including yet unknown types) would result in a compromise of ECU that can be recovered by a single method of ECU re-flashing. In other words, we were able to measure the resilience of the SUT to an infinitely large set of attacks, without testing or implementing any of the attacks individually. Instead, this allowed us to choose a particular ECU, determine what effects a compromise of that ECU would demonstrate, and then exhibit those effects via degraded performance by the SUT.

Note 4a. This step should start with identifying a few performance parameters of the SUT that are most relevant to the success of the business task (or a set of business tasks identified in Step 2) and are most likely to be impacted by the cyber attacks identified in Step 3. One source of candidate performance parameters can be the SUT documentation that often includes a list of Key Performance Parameters (KPPs). Consult SMEs for selecting most business task-relevant and attack-relevant parameters. Quantitative and binary parameters are strongly preferred. If a performance parameter is categorical, consider the possibility of using a numeric rating scale (e.g., a Likert scale), where a number is assigned (using objective, documented standards) to each category of performance.

For example, in the case study of a digital–physical model of a cargo truck, we have selected one performance parameter: fuel efficiency of the truck. It was of critical importance in case of the business task we selected (the truck's range was limited by the fuel on-board; there were no opportunities for refueling along the route; and if a cyber attack succeeded in decreasing the fuel efficiency, the business task of the truck could fail). It was also a parameter that could be strongly influenced by a successful cyber attack without necessarily creating a clear alert. In another case, SMEs determined that the SUTs ability to navigate to a business task-prescribed geographic area was the parameter most susceptible to cyber attack and highly detrimental to the business task success. We quantified the ability to navigate via the probability of successful arrival to the assigned area. In yet another case study,

SMEs selected a computer system response time as the business task-critical parameter most likely to be affected by the expected types of cyber attacks.

Note 4b. Once the test team identifies the individual performance parameters, the parameters should be aggregated into a single, aggregate performance measure. Numerous methods exist for such an aggregation. The most common method is a weighted sum of individual parameters, where the weight of each relevant parameter may be determined by SMEs and may depend on the goals of a particular business task (i.e., some business tasks depend critically on speed, while others succeed only by stealth). Appropriate weights can also be determined more automatically through analyses in which researchers repeatedly simulate business tasks for different values of each relevant performance parameter and then calculate the success rate under the given conditions. This procedure would be an application of what is known as expected utility theory, a commonly used normative framework in applied decision theory (e.g., Abbas and Cadenbach 2018).

Note 5. The SUT should be instrumented sufficiently to obtain the data that allow the test team to compute the parameters selected in Step 4. For example, in a case study of a digital–physical model of a cargo truck, instrumentation was provided to measure the amount of fuel consumed by the truck over time and the number of kilometers traversed by the truck over time. These two data items were sufficient to calculate the fuel efficiency (in kilometers per liter) of the SUT. This is done "on-the-fly" so that instantaneous fuel efficiency can be reported at regular time intervals across the business task duration. In practice, it is prudent to "over-instrument" the SUT so that additional data—assessed as potentially relevant—could be collected as well, in case the test determines that additional performance parameters must be considered.

Data collection should be able to provide data over time, from the beginning to the end of the business task. If the business task's entire time period cannot be covered, consider collecting data from a time prior to the start of the cyber attack and ending at the time when the aggregate performance measure reaches a steady state. Data can be physical in nature, such as fuel consumption and kilometers traveled. Data can also be strictly "cyber" in nature, such as the system response time or the volume of data exfiltrated by the adversary. In addition, data should provide insights on when the attack started, which subsystems it affected, and when the attack has been defeated or contained.

Note 6. The tools may range from manually operated scripts to a semiautomated or fully automated Adversary Emulator. Depending on the nature of SUT and on the nature of the attacks determined in Step 3a, the test team may be able to use one of the open-source or commercial tools for Automated Red Teaming or Adversary

Emulation. In other cases, custom tools may be needed, if the nature of the SUT and the attacks differs from what the available tools cover. For example, in the case of a cargo truck where we focused our attention on CAN bus attacks, available tools (at the time) could not provide the necessary functionality. Note the tools may emulate an attack or alternatively the effect of an attack (see Note 3b). The test team should determine which of these two approaches would be more effective for the purposes of measuring the resilience of the SUT. For example, in the case of the cargo truck, we implemented both approaches in order to provide both faster data generation and methodology testing cycles for the digital–physical model, as well as a means to test an actual cargo truck within a high-fidelity adversarial environment. Both approaches were implemented in a fashion that promotes experiment repeatability with a minimum amount of human interaction or intervention.

Note 7. Without cyber attacks and focusing on the KPPs identified in Step 4, execute a series of experiment runs with enough statistically relevant variability to produce a baseline of SUT performance. For example, in the case of a digital–physical model of a cargo truck, we added variability in the form of truck target speed randomization with a threshold of ±5 km per hour of the actual target speed in order to simulate driver attention drift. Every 1 to 10 s, a new target speed within the threshold would be chosen in increments of ±2 km per hour from the previous target. Prior to every run, a unique seed is used to initialize the pseudo-random number generator responsible for providing the target speed randomization so that each run varies in its simulated attention drift. Decide whether to capture all or only relevant performance data. Capturing all data and reducing via postprocessing is recommended so that captures can be reused in the future when new performance parameters have been identified. However, if the magnitude of data is large and experiment repeatability is not a concern, then capturing only relevant performance data will suffice.

Note 8. Repeat the same procedure as followed in Step 7 except with the inclusion of cyber attacks. If variability has been added, ensure that the randomization is initialized in the same manner in both the baseline (i.e., Step 7) and attack (i.e., Step 8) runs. Otherwise, performance comparisons will be inaccurate. Consider multiple runs with the same attack at varying attack strengths (if applicable), start times, and durations. This will provide keener insight into the degradation and recovery, if any, exhibited by the SUT.

Note 9. To compute the relative functionality of the system, simply divide the performance measure recorded in Step 8 (i.e., performance under attack) by the baseline performance measure recorded at the same time of the business task in

Step 7 (i.e., performance under normal business task condition, without a cyber attack). Do so for each time point of interest during the business task.

For the sake of illustration, consider Fig. 1. Suppose in Step 7 we measured the performance of the SUT (in this case the performance happens to be the fuel efficiency of the SUT) in baseline execution of the business task; we recoded it as the blue line in the upper panel of Fig. 1. Then, in Step 8 we measured the performance of the SUT while subjecting the system to a cyber attack. We recorded that level of performance as the orange curve in the upper panel of Fig. 1.

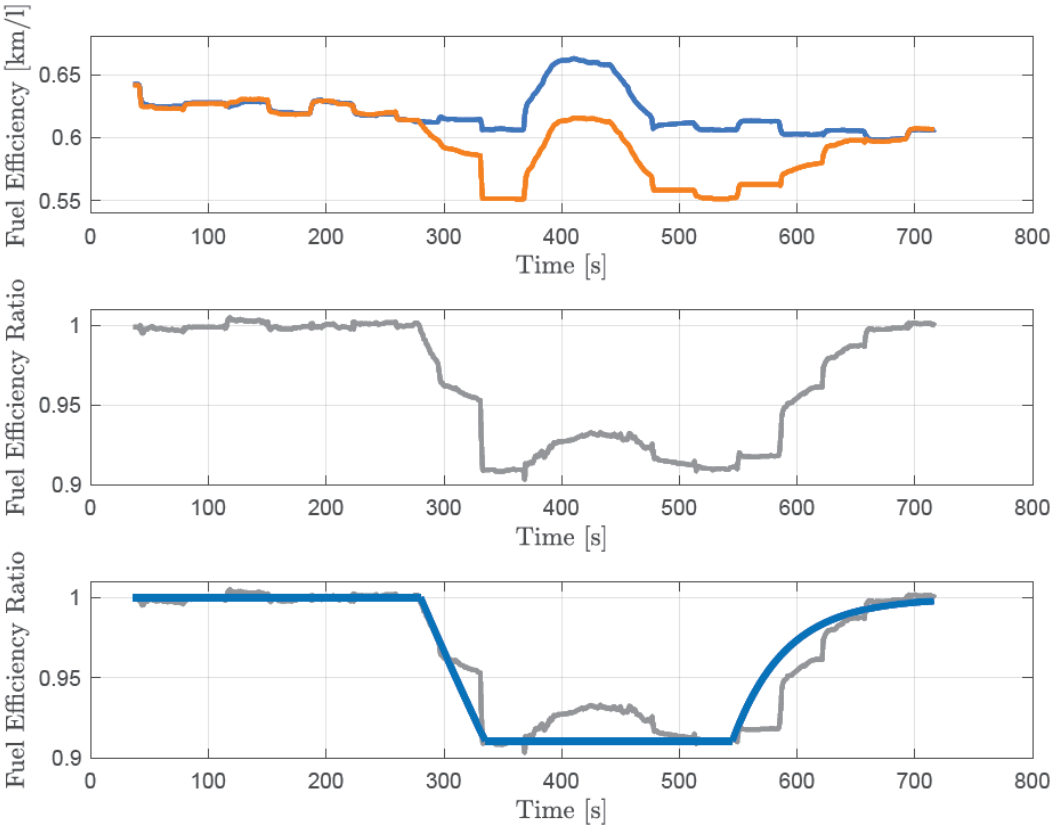

Fig. 1 Computing the relative functionality

Next, for each time point we divide the performance under attack (i.e., a value on the orange curve) by the baseline performance (i.e., the value on the blue curve). The resulting number is shown in the middle panel of Fig. 1 as the gray curve. This "bathtub" shape of the relative performance curve is typical: the value is close to 1.0 before the attack, then it drops, then stabilizes, and eventually recovers closer to 1.0.

Note 10. To compute resilience R, first decide on the period T over which you wish to define the resilience. There are two obvious choices, and for special cases you

might want to define a different time period. One choice is the period of the attack. Looking at Fig. 1, the attack starts to manifest itself f by a drop of performance at time 300, and recovery is completed by approximately 700. Therefore, the period of the attack is 400 s; we can use this as the basis for computing resilience, in the following manner. The area of the "bathtub" shape (i.e., the area limited by the line of 1.0 at the top and by the gray line below) is about 35 s. (You can use any numerical integration procedure to compute this area.) This is the amount of performance (dimensionless) lost over the period of attack, which is 400 s. The resilience is then R = 1 – 35/400 = 0.912.

Alternatively, you may select the entire duration of the business task (e.g., 10 h or 36,000 s) as the basis for computing resilience. In that case, the resilience R = 1 – 35/36,000 = 0.999.

In yet another approach, you might say that you expect such an attack to occur as often as every 10 min (i.e., 600 s), and you decide to use that time period as the basis for computing resilience. If so, the resilience R = 1 – 35/600 = 0.942.

An important point here is that when talking about resilience, we should specify which period of time we use for computing resilience.

Note 11. Experimental data also give us an opportunity to learn about the strengths of malware and "bonware" (everything that resists the malware) that participate in the process of losing and then recovering the performance of SUT. For details on definitions and on rigorous approaches to computing the effectiveness of malware M, and the effectiveness of bonware B, see the Appendix. Here, we describe a simplified procedure that assumes we can approximate the resilience episode (i.e., an attack followed by a recovery) with a bathtub shape illustrated in Fig. 1.

In Fig. 1 (middle and lower panels), we see that malware causes a rapid drop in SUT's relative functionality starting at time, t1 = 280, when the relative functionality is 1.0 (i.e., F1 = 1.0), and ending at time, t2 = 335, when the relative functionality is diminished to approximately F2 = 0.9107. Assuming the bonware is not yet active in this period, we can compute the effectiveness of malware M using the following formula:

$$M = \ln(F1/F2)/(t2–t1), \quad (1)$$

where ln is natural logarithm. If you use Excel, use function LN.

In this example we obtain M = ln(1/0.9107)/(335–280) = 0.0017.

Then, at time approximately t3 = 545 the relative functionality starts to recover from F3 = 0.9107 and reaches the value of F4 = 0.998 by the time t4 = 716.

We assume here that the malware is no longer active in this period. In that case, we can compute the effectiveness of bonware B using Eq. 2:

$$B = \ln((F3–F1)/(F4–F1))/(t4–t3) \quad (2)$$

In this example, we obtain

$$B = \ln((0.9107–1)/(0.998–1))/(716–545) = 0.0222. \quad (3)$$

Note 12. Any value of a quantitative measurement is meaningful only in context, particularly in comparison with other values. In our case, a value of cyber resilience is meaningful only in comparison with values measured for comparable systems. For example, it is difficult to decide whether a value R = 0.912 is low or high. On the other hand, suppose a truck without an autonomous cyber recovery module exhibits R = 0.63, and with addition of such a module it exhibits R = 0.912. Then we might conclude that introduction of the additional module results in a major increase of cyber resilience.

4. Conclusions and Recommendations

We assess that the proposed methodology exhibits the following features:

- supports the ability to execute a diverse series of experiments and collect detailed data;
- supports the ability to compute AUC-based resilience measure from experimental data;
- supports the ability to derive newly proposed efficiency coefficients for malware and bonware; and
- produces experimental results that are physically explainable, adequately stable, and meet monotonicity expectations.

We make the following recommendations:

- Application of the QMOCR methodology is appropriate—and should be considered—when an actual system, its prototype, or a working, executable model (physical, digital, or digital–physical) is available. In that case, the methodology uses such a *working* system or model for quantitative, experimental comparison of the system's behaviors in normal operation and under attack.
- In some cases, a thought experiment (i.e., tabletop experiment) may be performed when the participants of the thought experiment have prior

experiences observing the behavior of the system/model in operation or tests.

- On the other hand, the QMOCR methodology is *not* appropriate at those phases of system design and development when a working prototype or working model are not yet available. In such cases, it may be appropriate to perform a structured qualitative assessment (not described in this report) by SMEs using *descriptive* models of the system under development (drawings, diagrams, schematics, process flows, formal specifications, functional decompositions, etc.).

5. References

Abbas AE, Cadenbach AH. On the use of utility theory in engineering design. IEEE Sys J. 2018 June;12(2):1129–1138. doi: 10.1109/JSYST.2016.2602562.

Alexeev A, Henshel D, Levitt K, McDaniel P, Rivera B, Templeton S, Weisman M. Constructing a science of cyber-resilience for military systems. NATO IST-153 Workshop on Cyber Resilience; 2017. p. 23–25.

Beling P, Horowitz B, McDermott T. Developmental Test and Evaluation (DTE&A) and cyberattack resilient systems. TR SERC-2021-TR-015 (V2). 2021 Sep.

Kott A, Linkov I. Cyber resilience of systems and networks. Springer International Publishing; 2019.

Kott A, Linkov I. To improve cyber resilience, measure it. Computer. 2021 Feb;54(2):80–85.

Linkov I, Trump BD, Keisler J. Risk and resilience must be independently managed. Nature. 2018;555:7694.

Linkov I, Eisenberg DA, Plourde K, Seager TP, Allen J, Kott A. Resilience metrics for cyber systems. Environ Syst Decis. 2013;33(4):471–476.

Smith SC. Towards a scientific definition of cyber resilience. In: Proceedings of the 18th International Conference on Cyber Warfare and Security (ICCWS 2023); 2023; Red Hook, NY. p. 1–9. Academic Conferences Ltd.

Appendix. Quantitative Measurement of Cyber Resilience: Modeling and Experimentation

The following Appendix is a technical paper that defines key concepts used in this report, describes an example of an experimental test-bed, and an experimental technique that can serve as a simple example of applying the QMOCR methodology. It also discusses the mathematical techniques used to process the experimental data within the QMOCR methodology.

List of Symbols, Abbreviations, and Acronyms

3-D	three-dimensional
ARL	Army Research Laboratory
CAN	controller area network
DEVCOM	US Army Combat Capabilities Development Command
ECU	engine control unit
KPP	Key Performance Parameter
QMOCR	Quantitative Measurement of Cyber Resilience
SME	subject-matter expert
SUT	system under test

1 (PDF)	DEFENSE TECHNICAL INFORMATION CTR DTIC OCA
1 (PDF)	DEVCOM ARL FCDD RLB CI TECH LIB
5 (PDF)	DEVCOM ARL FCDD RLD A KOTT FCDD RLA ND MJ WEISMAN JE ELLIS TW PARKER SC SMITH