Resilience is More than Availability

Matt Bishop
Dept. of Computer Science
University of California at Davis
Davis, CA 95616-8562
bishop@cs.ucdavis.edu

Marco Carvalho
Institute for Human Machine Cognition
15 SE Osceola Ave
Ocala, FL 34471
mcarvalho@ihmc.us

Richard Ford
The Harris Institute for Assured Information
Florida Institute of Technology
150 W. University Blvd
Melbourne, FL 32901
rford@fit.edu

Liam M. Mayron
The Harris Institute for Assured Information
Florida Institute of Technology
150 W. University Blvd
Melbourne, FL 32901
lmayron@fit.edu

ABSTRACT

In applied sciences there is a tendency to rely on terminology that is either ill-defined or applied inconsistently across areas of research and application domains. Examples in information assurance include the terms resilience, robustness and survivability, where there can exist subtle shades of meaning that differ between researchers. These nuances can result in confusion and misinterpretations of goals and results, hampering communication and complicating collaboration. In this paper, we explore the ideas of resilience, robustness and survivability, and propose security-related definitions for these terms. Using the proposed terminology, we then argue that research in these areas must consider the functionality of the system holistically, beginning with a careful examination of what we actually want the system to do. Based on the literature, we note that much of the published research focuses on a single aspect of a system – availability – as opposed to the system's ability to complete its function without disclosing confidential information or, to a lesser extent, with the correct output. Finally, we discuss ways in which researchers can explore resilience with respect to integrity, availability *and* confidentiality, instead of just availability.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous; D.2.8 [**Software Engineering**]: Metrics—*complexity measures, performance measures*

1. INTRODUCTION

Resilience, robustness. . . what's in a name? Resilience is a desirable property for modern computers, and we tend to use the term interchangeably with robustness, even though in practice these properties are quite different. A fundamental step towards understanding, building or enabling these properties in man-made systems is to define and measure them clearly in a way that is meaningful and widely-accepted.

In this paper, we seek to clarify what we mean when we write "attack resilience" in the context of computational systems and infrastructures. In addition, we attempt to highlight some of the different dimensions of the problem, including distinctions between the capabilities and properties of a robust system compared to those of a resilient system. Our goal is to provide a framework for discussion that will help the community to converge on terminology, and better relate its intended objectives and metrics associated with systems design, implementation and validation.

We often visualize a system providing a service and interpret the resilience of the system in the context of that service. A "resilient" network routing infrastructure, for example, is expected to continue operating at or above minimum service levels, even under localized failures, disruptions or attacks. Continuing operation, in this example, refers to the service provided by the communications infrastructure (or its mission, in military parlance). If, despite local failure or attacks, the routing infrastructure is capable of maintaining its core purpose (routing packets) we call it robust, or maybe survivable, or possibly resilient, depending on who you talk to. However, these terms are not synonyms — each has subtle shades of meaning. We argue that such looseness of terminology is dangerous, especially when examining problems which have a strong interdisciplinary component. To this end, it makes sense for us, as a community, to collect our thoughts about these topics in an ordered way.

Very often, the concepts of resilience, robustness and survivability are more easily described from the perspective of systems performance, as this is more readily measured and controlled than more abstract qualities such as system confidentiality. However, when performance must be balanced with confidentiality and integrity requirements, the concept of a resilient system starts to become fuzzy. For example, in our routing example above, one could imagine a scenario where, if local CPU load were too high, the system might dispense with end-to-end encryption, trading confidentiality for availability. Clearly, such an act has impacted some dimensions of the system related to its resilience and robustness, but we currently lack the terms and world view to describe this trade-off crisply.

The remainder of this paper is as follows. We first highlight the importance of consistent terminology, and offer some definitions of terms that we believe are helpful in highlighting the similarities and differences between several related concepts. We then provide a brief literature survey which shows some of the different ways the term "resilience" has been used with respect to computer operations. Finally, we present a roadmap of how our proposed definitions can be used to explore the concept of resilience not only for availability, but also for confidentiality and integrity.

2. THE POWER OF WORDS

According to the principle of "linguistic relativity" words and thought are intimately linked. The Saphir-Whorf hypothesis, for example, argues that our cognition is constrained by our patterns of language [27]. While this approach has been subject to criticism, most modern scholars would agree that even if language does not direct cognition it most certainly influences it [2]. As such, the words which we use to describe a problem shape our conception of it (and vice versa). Furthermore, if the words we use are imprecise, there is a strong chance that our thoughts are equally so. Words have, perhaps, more power than we realize.

Before exploring these definitions, we note the overall lack of metrics in the field of computer security. Even if we were to agree that, in the loosest possible sense, a robust system is one which does "well" when under attack, it would be very difficult to measure what we mean by the term "well". With this in mind, it is interesting to look at the raft of words we have to describe the loose area of reliability.

2.1 Qualitative Definitions

The terms resilience, robustness and survivability are all words that are used outside the technical literature. As such, their original "plain English" meanings and associated connotations are likely to color the way in which these words are perceived, even when used in their technical sense. We therefore begin our discussion with the non-technical definition of these terms. Despite common technical usage and redefinitions, the words carry with them much of their original connotations; to ignore this is naive.

In common language, *resilient* is either "resuming the original shape or position after being bent, compressed, or stretched" or "rising readily again after being depressed; hence, cheerful, buoyant, exuberant" [22]. Thus, when we think of a system being resilient to an attack, we are stressing the ability of the system to recover from the impacts of this attack, or at least to maintain the potential of autonomous recovery. It is important to note that such capability implies that the system must not cease to exist—that is, it must survive at some capacity, in order to autonomously recover. In the cyber domain, a resilient system continues to provide essential functionality, even under duress or in an impaired state. Figure 1 shows this pictorially.

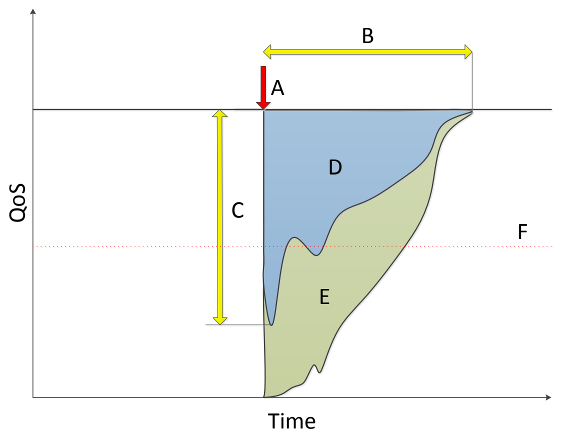

Figure 1: Possible responses of a system to an impulse at time A. B represents the time taken for the system to return to its equilibrium Quality of Service (QoS). C represents the maximum disturbance for system D. Another possible response is shown for system E. Finally, line F represents a QoS below which the system's mission is compromised.

By way of contrast, *robust* is defined as "strong and hardy in body or constitution; possessed of rude strength; strongly and stoutly build; of a full and healthy habit" and "strong, vigorous, healthy" [22]. Thus, a robust system is one that is sturdy. The system can still fail, but it is less likely to transition to a partially-compromised state. It will either fail completely or not at all, and by robustness alone it will remain in this new state until someone or something intervenes.

Survivable, defined as "capable of surviving," comes from the word survive, which is defined as "to continue to exist after some person, thing, or event; to last on" and "to continue to live after (an event, point of time, etc.), or after the end or cessation of (a condition, etc.)" [22]. A survivable system has the capacity to continue to exist even under highly degraded operational conditions. For all intents and purposes, one may argue that a system unable to operate at or above minimum Quality of Service (QoS) requirements has technically failed to survive, which is true for mission critical systems and applications. The Internet is often cited as a good example of a survivable system, where very large attacks may significantly disrupt its functionality, but will be very unlikely to completely disrupt its operation. Graceful degradation is a characteristic of survivability.

On the surface, resilience, robustness, and survivability seem to be three different capabilities. A deeper analysis reveals that they all contribute to different aspects of a system's health and defense. They provide an orthogonal perspective to the important concept of system reliability, which is also often confounded and misused in technical discussions.

Reliability is "the quality of being reliable, reliableness" [22]. Following this to its root, we find that reliable is closely related to dependable—the idea of being able to rely on the system. This is more formally given by Randell et. al. [18], who write "[t]he reliability of a system is taken to be a measure of the success with which the system conforms to some authoritative specification of its behavior." They go on to point out that we can measure *aspects* of reliability; reliability as a whole is dependent on context.

A system that is easy to degrade but nearly impossible to impair completely may be intrinsically survivable, while a system that is difficult to degrade under a wide range of attacks and operational conditions (or maybe not all) is generally accepted as robust. Resilience, on the other hand, brings in a new dimension. Not only are resilient systems expected to maintain their operation under attack or failures, but they are also expected to remain mission-capable, that is, to reconfigure or recover in order to restore its original state. However, even resilience requires survivability to flourish, as a brittle system will likely fail completely before it has an opportunity to restore itself. Resilience is commonly found in natural systems, and is a hallmark of the biological world.

Going back to the operational levels defined as part of our QoS and system capacity discussions, one could argue that a robust system operates at or above desired QoS requirements, while it may progressively consume its reserves until it fails catastrophically. A survivable system may have loss of capacity and service degradation, possibly operating below desired levels, but still above minimum QoS requirements; and a resilient system is effectively a survivable system that is capable of restoring not only its performance level back to desired levels, but also the capacity of the system itself to recover, maintaining its ability to sustain future attacks or failures.

At the highest level, we propose that a resilient computer will recover and return to "what it was doing" before an attack. However, in the context of security, it is not that simple. Computer security is not just about availability (though all too frequently, that is what we actually mean when we talk about resilience), but also confidentiality and integrity. Unfortunately, resilience in these dimensions is much more challenging. In terms of confidentiality, once your data has been made public, recovery seems rather difficult (but in fact, all is not lost—we have chosen our words *very* carefully here). While there are certain secrets that can be painlessly replaced *in situ* (for example, if we discover a session key has been disclosed, we could negotiate a new one on the fly), resilience in general with respect to confidentiality seems like a tough problem, yet it is clearly a large component of attack resilience. Similarly, integrity is a tricky thing; data does not generally sit around at rest—the purpose of it is to move about and be leveraged, allowing users to draw conclusions. Here again, there has not been enough discussion of recovery, except when we consider things like backups and version control; these concepts must exist in a computer system to be truly resilient.

Instead of formal definitions of the terms, we offer the following table that provides insight into the nuances of each term. These nuances represent the way in which we view these terms, and will apply to them for the rest of the paper.

Word	Nuances	Example
Resilient	Implies system may degrade, but will recover.	Grass under load bends, but when the load is removed, it springs back in to shape.
Robust	Resists deformation; sturdy. Does not imply an ability to restore lost function.	A brick building, in a storm resists the wind; however, any damage is permanent until someone intervenes.
Survivable	A higher level property—in some sense the system will survive. Has some implication of recovery.	A large distributed system, or routing infrastructure

Table 1: Terms and their shades of meaning, plus examples.

3. A BRIEF HISTORY OF RESILIENCE

One of the foundational concepts we are arguing for is presented in Ellison et al's "Survivable Network Systems" [6], where the authors highlight a critical concept of survivability as maintaining "essential properties, such as integrity, confidentiality and performance". The authors further define survivability as "the capability of a system to fulfill its mission, in a timely manner, in the presence of attacks, failures, or accidents." In addition, the authors argue for four fundamental properties: resistance to attacks (what we would now consider to be robustness), recognition of attacks and the extent of damage, recovery of *full* [our emphasis] and essential services after an attack, and adaptation and evolution to future attacks.

With this as a grand vision, it is interesting to see how real-world research into survivability has developed. Part of the literature has focused on the concept of fault tolerance. Here, much has been made of topics like n-version programming and redundancy. These approaches are effective, but provide only part of the system we desire. For example, RAID protects us from hardware failure, but is powerless to protect us from a software error that writes incorrect data to a database. In essence, we have worked hard at the component level but need to turn our attention away from hardware failure and toward the entire system in a more holistic approach. A similar story can be told for parity bits and error correction systems, though again, these approaches are most applicable to unintentional error rather than coordinated attack. Forward Error Correction and Backward Error Correction are almost a form of redundancy, allowing either retransmission or correction of transmitted data. However, these approaches are narrow and concern themselves primarily with data in motion—they do not account for bad data derived from the source. At a very low level, these technologies are helpful, but form just part of a complete solution.

Part of the challenge we face is that there is a vast difference between considering accidental error (a bug literally crawling into a machine and causing a short circuit) and intentional error. Redundant systems protect from hardware failure, but if they represent a monoculture in terms of program logic or implementation, they can be brought down quickly and easily by an attacker. Thus, much work in fault tolerance is of limited value to the security practitioner, unless some source of diversity is introduced. Sources of diversity could be natural (n-version programming [5]) or synthetic (compiler-generated diversity [8]), at which point we have at least limited the single point of failure.

For better or for worse, the term "resilience" is not always used in a consistent way. In "Enabling Distributed Security in Cyberspace" [19], a recent publication by the Department of Homeland Security (DHS), we read the following definition:

> Resilient. For cyber defense purposes, having sufficient capacity to simultaneously collect or receive and assess security information, execute any ACOA [Automated Courses of Action] make alterations to the ACOA as needed, and sustain agreed upon service levels.

Note that this definition does not really address the idea of recovery (except perhaps as a side effect of altering the ACOA)—it fits more under the heading of robustness. A system that dips below an agreed-upon service level is not, per this document, resilient. Similarly, Trivedi et al. write: "In general, resilience can be defined as the ability of (system/person/organization) to recover/defy/resist from any shock, insult, or disturbance" [24]. We argue this definition is sufficiently broad as to render itself of limited use. To conduct science, a narrower and more precise agreement on terminology is needed.

From an organizational resilience perspective, Woods [28] introduces the concept of resilience engineering, and highlights the difference between the resilience of a system and its ability to absorb or adapt to disturbance. Resilience, as defined in that context, concerns the ability to recognize and adapt to unanticipated situations outside the competence envelope of the organization, demanding a shift in process, strategy and organization.

As we can see, the terms we use are often defined differently among papers. Survivability, robustness, and resilience are used interchangeably within individual papers, and we often lose both clarity and meaning when discussing these issues.

Part of the challenge is that we suffer from a lack of metrics. In the absence of a consistent definition, system properties we measure can provide a format for at least creating "apples to apples" comparisons even if we are unsure how to name the property we are measuring. Some work has been done in this area, but it has problems. For example, in [10], we read about a property called "system reliability" that measures the probability of time to failure within a certain window. The reasoning is that this measure tells us, for different values of t, the utility of the system for calculations that take different times to complete. The authors propose another metric, "service reliability," that considers the conditional probability that the system will be able to complete a particular transaction given a particular job. However, these metrics seem better suited to random error than coordinated attack.

4. CAN WE QUANTIFY RESILIENCE?

For the purposes of this discussion, we assume that robustness, survivability and resilience are defined for an operational system providing a well defined set of services. Associated with such services are requirements that define quality of service (QoS) metrics.

Such QoS metrics are typically thought of as defining performance and availability, but in fact there is no *a priori* reason that they could not measure aspects of confidentiality and integrity. While we typically think of a system as preserving (or not preserving) integrity and confidentiality, there are actually degrees of preservation. It may be difficult to put a number on the confidentiality provided by a system, but all other factors being equal, we can say that a system which uses a 256-bit key is providing "more" confidentiality than a system with a 16-bit key.

The relative ease with which we can measure typical QoS parameters, combined with our tendency to think of security as a discrete property (that is, a system is either secure or it is not) have also shaped our experiments and publications. As computer scientists, we think of data in its binary representation; thus, data is either right or wrong — integrity is preserved or it is not. In contrast, a signal analysis approach might attempt to measure the distortion of a waveform against its actual value, leading to a ranking of systems in terms of the fidelity they preserve.

This discrete "yes or no" view of the world has served us well in some senses, but may be a limiting factor in our ability to quantify and reason about resilience with respect to integrity. Approaches which take a less black and white approach to data integrity do exist. For example, some measures of data integrity attempt to take a probabilistic approach [14]. However, work in this area is rather scant.

The lack of base metrics for integrity and confidentiality are exacerbated by the nature of resilience—indeed, other disciplines have struggled with this problem alone. Tierney and Bruneau [23] try to define and measure resilience with respect to disaster loss. They introduce the R4 framework of resilience:

- Robustness—the ability of systems, system elements, and other units of analysis to withstand disaster forces without significant degradation or loss of performance;

- Redundancy—the extent to which systems, system elements, or other units are substitutable, that is, capable of satisfying functional requirements, if significant degradation or loss of functionality occurs;

- Resourcefulness—the ability to diagnose and prioritize problems and to initiate solutions by identifying and mobilizing material, monetary, informational, technological, and human resources; and

- Rapidity—the capacity to restore functionality in a timely way, containing losses and avoiding disruptions.

The paper concludes that resilience could be measured by examining these component parts. However, it does not go further.

Ives [12] argues that resilience in ecology can be measured by looking at the time it takes the system to return to its equilibrium state after a perturbation. This measure, combined with the *resistance* of the system (the magnitude of change to a particular stimulus) appropriately characterizes the system[1].

Other domains in which measurements have been proposed include economics [3], where an index is proposed to estimate the policy-based ability of an economy to withstand

[1]This approach is only partially correct as it is easy to draw curves that have the same magnitude and recovery time, but that have very different operational qualities

and recover from exogenous shocks. In industrial control systems [20, 26] some authors have estimated resilience as a function of performance loss, based on the time a system can withstand an attack, and the time it takes to recover. Other metrics [11, 16] have also included estimates of system capacity, operational limits (margin), tolerance and flexibility/stiffness.

Combined, these metrics enable the specification of desirable and minimum levels of service for the system. Desirable QoS defines the normal operation conditions of the system, while minimum QoS defines the lowest levels of service necessary to ensure a successful, although possibly degraded, service execution. A system whose performance is degrading will operate at progressively lower levels of QoS until it crosses its minimum QoS requirements, at which point it may still be operational, but it has failed to maintain service continuity. Thus, an operational system may be functioning at or above desired level, below desired levels but still at or above minimum levels, below minimum levels, or not operating at all, at which point it has been completely disrupted.

By analogy, we can think of the reserves of the system as potential energy, and the function provided as kinetic energy. As the system deals with failures, potential energy is traded for kinetic, keeping the system "in motion". Thus, it is meaningful when considering system health to consider not just how it is performing, but also what it has in reserve.

The reserves of a system define its ability to maintain its operation at the required levels of service (which may include measures of performance/availability, integrity or confidentiality). Notice that system reserves are different from, but highly related to, QoS maintenance. A drop in system reserves does not necessarily affect any of the QoS metrics, but it may affect the ability of the system to ensure the maintenance of such metrics in the future.

By design, we often build fault-tolerant systems with redundant capabilities to ensure service continuity against isolated failures. For such systems, localized failures may disrupt redundant copies of different components without necessarily compromising the continuity of its service. For each loss, the system appears to "fight through" from the perspective of an outside observer. The ability of the system to cope with further disruptions or failures is reduced however, and such losses are neither automatically recoverable or infinitely sustainable; as a function of time, the ability of the system to absorb further loss is reduced. By observation, it is more susceptible to failure, though this change is hard to quantify, as no QoS reduction may be observable or measurable.

Thus, from a systems perspective, a reliable system is one that reliably, or consistently, provides the expected response within the expected time to a given query or service request. Note that the expected response is not necessarily the correct one, but it is one that the programmer or designer anticipated – it may be deeply suboptimal in practice (for example, a Web server under attack will typically continue to serve web pages to the attacker – this is correct with respect to the design, but hardly correct in the larger picture). Thus, there is a need to identify and better define system properties for higher level behaviors associated with service continuity, including concepts that goes beyond the notion of reliability.

Aside from the challenge of measuring resilience (i.e. recovery from attack), we are also severely hampered by a lack of metrics for integrity and confidentiality. As these properties are highly situational, it seems unlikely that any universal approach will be developed soon. However, at least with respect to a particular attack on a particular system, it should be possible to compare the resilience provided by different protection mechanisms qualitatively.

5. A PARADIGM REVISITED

Our new paradigm is to move toward resilience with respect not only to performance and availability but also to confidentiality and integrity at the higher levels. This is not to say that robustness is not always desirable—many of our traditional computer architectures attempt to provide these qualities (for example, checksums allow us to detect and delete corrupted packets, encryption makes our traffic robust to eavesdropping in transit, and automatic failover lets us accept the failure of a single machine). In addition, we believe there are significant opportunities for work that provides resilience (that is, essentially, recovery) at the higher levels. Furthermore, some level of combined robustness and resilience is possible in both the confidentiality and integrity domains.

Viewing the system holistically is critical when we consider attack resilience, as is viewing the goal of the attacker more widely. Solutions have to be broader than simply focusing on service provisioning regardless of the operational conditions. They must also provide service while maintaining other significant aspects of service expectations, such as keeping a transaction confidential while still providing some level of attribution and non-repudiation. Ellison's technical report [6] directly requested this, but the concepts seem to have been sidelined at best, and ignored at worst, by the mainstream security researchers.

The nature of resilience changes depending on the problem being addressed. If a computer system were tasked with producing a simple "yes" or "no" answer to a question, there is no "good enough" performance. Such a system will either work or it will not, and the best we can hope for is an acceptable error rate (which will also vary depending on context). In contrast, a system designed to control the temperature of a room or control a bipedal robot can work acceptably with inexact solutions. Therefore, an essential part of building resilient systems is thinking about the problem at hand and recasting it from a discrete (and hence brittle) problem to an analog problem. This alone opens the door to solutions that provide the benefits of information technology without its inherent tendency toward failure under unexpected conditions. Furthermore, depending on attacker motive, a more analog view of the world may allow us to protect the confidentiality of data even in the presence of a breach.

Resilience for availability, integrity and confidentiality ultimately refer to the overall goal of the system. Complex systems are highly interdependent and the effects and implications of these properties are not necessarily trivial. We propose that resilience, robustness and survivability are systems-wide properties, not necessarily achievable with the individual specification of component properties and requirements. Local violations of these properties at the component level may not only be acceptable but sometimes necessarily for the resilience of the overall system.

6. RESILIENCE FOR INTEGRITY: A ROAD MAP

Integrity looks like a difficult problem from a resilience standpoint because how to recover from incorrect or corrupted data is not immediately obvious. Much work has looked at replicating data and then "voting" on the "correct" copy of it. Unfortunately, these approaches sometimes force the user to trade confidentiality risk for integrity risk. While this trade-off seems natural, other approaches do not require systems to place data at risk, such as distributed backup solutions. Even here, though, the data is at risk from those who hold the decryption keys.

Sadly, confining our concept of integrity to stored data is an oversimplification. Often, we care far less about the input to a calculation than its output. For example, if an industrial control system is controlling a process well, a particular sensor providing faulty data is of limited concern. From a purely pragmatic perspective, we note that our concern is not with the internal machinations of a process, but the utility of its output. Thus, while our current programming paradigm tolerates the "GIGO" mantra (garbage in, garbage out) natural systems spend considerable effort sanity checking and correlating often conflicting sources of information to make a "best effort" attempt at satisfactory output.

Inexact but "good enough" is in fact a promising way of dealing with approximate or erroneous data. Organisms often use these approaches. When balancing a check book, for example, if one's balance based on addition and subtraction is radically different than the expected value, a rational person will assume that the calculation is flawed either due to error or incorrect input data. Essentially, a flag is raised, and the human engages in recovery, perhaps by rerunning the calculation or checking, line by line, each input. This is based in part upon prediction using an algorithm that gives us a rough approximation of output with limited calculation.

Typically, our computations on computers do not work this way. A single thread takes input and produces output with no expectation of magnitude. Information is typically treated as utterly devoid of context. Developing algorithms with a predictive component that can detect error and then contextualize the next step is crucial. Here, we believe biological systems are of significant interest in terms of design. For example, Mayron et al. [15] explore the question of symptom representation and interoception and outline a system that attempts to "understand" symptoms. Biological systems—especially those that look at symptom representation and comprehension—are particularly good at dealing with conflicting information. This model avoids accepting input as true and correct and instead focuses on combining different and potentially conflicting data sources into an overall world view that can then be tested. While this does not make the input data itself resilient with respect to faulty input, it *does* protect the process from bad input and allows it to recover. This focus shift is important, because it allows us to contextualize and reason about calculations instead of simply act on them.

In addition, the types of activities that biological systems participate in are rather different than those we have computers to carry out. Consider the challenge of catching a ball. The object must be identified and its trajectory calculated. As the ball approaches, our approximate solutions get closer and closer to reality as the quality of our information improves and as we receive feedback on our movements. The solution space is approximate, analog, and forgiving of errors (provided that as a whole the system actually *works*). Conversely, if a computer is tasked with calculating the standard deviation of a set of numbers there is no feedback or approximation—with a specified degree of numeric precision, the answer is either right or wrong. This is a very different problem space, with very different constraints. It is interesting to note that biological systems (for example, graduate students) often provide wildly inaccurate answers to these kind of questions; different types of problem will require different mechanisms for resilience. We should be wary of drawing too many parallels with biology when we are operating in a very different domain.

In part, the primarily linear (and singular) execution of instructions has driven our approach to computing. Our programming languages are actually predicated on everything working *perfectly* and our algorithms typically account poorly for conflicting data. As an individual machine is typically subject to subversion by an adversary who can completely control its execution environment, it seems clear that we cannot rely on single machines to create a truly resilient system.

We argue that at least a two-fold approach is needed here. Individual machines should be able to check that outputs obtained make *sense* in context. Also, machines must be physically and logically separated so that a single subverted computational device cannot produce a (holistically) damanging result.

Previous approaches have focused on the latter part of the problem rather than the former. For example, in n-version programming, separate teams of programmers work on the same problem and create different solutions. These solutions vote based on output, and the "winning" solution is chosen. The idea is that the "independence of programming efforts will greatly reduce the probability of identical software faults occurring in two or more versions of the program" [5]. While this is true, it does not consider GIGO-related problems. If incorrect data is supplied to all participants, the system will still return a wrong (but numerically correct) result. As noted in Mayron et al. [15], the lack of context for information received impedes both resilience and robustness. A biological system typically uses a fast and frugal assessment of state before embarking on a calculation. If the assessment disagrees with the calculation by a large amount, more introspection is needed.

In addition to deployment issues, there is also room for creation of algorithms that are more tolerant of erroneous data. Swarm-based systems have been explored in this area (Bonabeau et al. [1] provide an excellent overview; Jiang and Baras present a specific application [13]). Resilience is considered at the higher level of the system, not at the individual swarm member, and is often tied to the elasticity of individual elements. A related approach uses the notion of "organic resilience" [4], in which localized state estimation in distributed systems allows for components to individually assess the response of other components in the system. The field is broad, and crosses multiple disciplines. Silos of work exist—some of it very good—but an integrative approach that pulls it together into a larger vision of resilient systems is lacking. We challenge the security community to lead the charge, and re-imagine our network systems. More robustness does not equate to resilience—a future-looking system has to incorporate aspects of both, and computationally may look very different to the deterministic approaches we are comfortable with today. These approaches often provide solutions that are near optimal, as opposed to exact.

7. RESILIENCE FOR CONFIDENTIALITY: A ROAD MAP

Given our desire to recover from a successful attack as opposed to simply resist one, we must pay more attention to the ways in which systems handle decision making and, more broadly, data. Shannon's theory of information tells us that information transmission is about uncertainty (and the lack thereof) [21]; grasping this allows for the possibility of novel approaches to resilience with respect to confidentiality. For example, a password of one of the authors is cleverly hidden in this manuscript, but as the secret is buried in a sea of otherwise irrelevant data, an attacker gets little benefit from this, as the uncertainty regarding the password is not significantly reduced.

Shannon and Weaver's book [25] illustrate this concept. The authors make a distinction between different levels of communication. They define these levels as [25, p. 4]:

- Level A. How accurately can the symbols of communication be transmitted? (The technical problem)
- Level B. How precisely do the transmitted symbols convey the desired meaning? (The semantic problem)
- Level C. How effectively does the received meaning affect the conduct in the desired way? (The effectiveness problem)

Our belief is that by leveraging Levels B and C of this hierarchy, it should be possible to create systems that provide some resilience with respect to confidentiality by undermining the semantics of communication, even though the actual bits which make up our private information (at Level A) have been transmitted.

By way of illustration, we have all seen attacks that chip away at a system by reducing uncertainty about content–essentially leaking information with each query response. When probing a system, uncertainty is key to resilience in this area. For example, Neagoe and Bishop [17] propose an excellent method for resilience with respect to confidentiality, based on the concept of actionable data. By mixing the false with the true, the actual information imparted may approach arbitrarily close to zero. This is illustrated below.

Consider a data leak prevention system that detects a user in the process of exporting certain confidential data from a company. Instead of shutting down the link, the system injects erroneous information into the stream. If the attacker has no way, *ex post facto*, of determining when this injection process began, which data were correct, and which were not, little information has actually been transmitted at Weaver's Level B & C because the transmission has done little to reduce uncertainty on the part of the attacker. Clearly, whether this is an acceptable approach depends on attacker's motive and the type of data being exported. Just because 1's and 0's are exiting a network correctly, it does not follow that we are operating effectively at the semantic level. We note that this approach would work because while the system's internal behavior changes when the exfiltration is discovered, the attacker does not know when. This effectively reduces the information content in the data flowing from the system. If we had merely terminated the flow, the attacker would know that the data extracted were correct, and potentially could infer far more usable conclusions from it. The perfect example of this is export of data to Wikileaks. If an exfiltration attempt were to be detected, it might make more sense to salt the true data with believable nonsense than to cut the flow in some cases. After the fact, the leaked data can be shown to contain inconsistencies, making it difficult for analysts to infer too much from the leak in the absence of external corroboration.

Whereas biology is often a good source for inspiration, we have struggled somewhat with finding biological inspiration for issues related to resilience of confidentiality (though sociology does provide potential research avenues). However, despite their scarcity, some ideas do translate well.

One particularly promising approach to confidentiality is the concept of negative databases [7], where we store the information we don't care about, allowing us to make inferences about information we do care about. Conversely, methods where the actual data we care about is present, but is spread about many places are also quite attractive. Recent breakthroughs in cryptography also make possible solutions where systems can draw inferences from data without any individual system having the raw data we wish to protect [9] – this increases our robustness. Like in nature, it is the opportunistic and emergent combination of many of these approaches and capabilities that will allow new computation systems to autonomously resist, survive and recover from attacks and failures with respect to information disclosure.

A final pitfall here is that use of a game-theoretic metaphor regarding uncertainty will only make sense when considering an attacker whose motivations can be correctly quantified. For example, in some circumstances, uncertainty about outcome is not much of a preventative. If, for example, we know a certain list of names contains one person who is a threat to us, we may move against *all* of the names on the list. Uncertainty of accuracy is protective only in certain cases.

8. CLOSING THOUGHTS

One of the issues we encounter in security is that a *mostly* secure approach is not good enough. This tendency to reject solutions is a huge issue for those working in the resiliency space. Despite attacker adaptation, if an approach removes options from an attacker, it is an incremental step toward a truly resilient system. We argue that systems that provide incomplete protection are not necessarily evolutionary dead ends, and we would do well to explore why our adoption curve seems to follow what is *hot* as opposed to what is proven, at least within certain bounds.

In the process of writing this paper, we have identified many papers on resilience and related topics, yet when looking at "real world" systems, the adoption of these techniques is very low. Where we have used technology to stave off disaster, we are often trading availability or integrity for confidentiality. Solutions that provide more resiliency seem to be economically impractical. Our sense is that truly resilient computer systems are possible, but will not be adopted any time soon for pragmatic purposes, despite the high-value assets computers control. History speaks pretty loudly about our actual desire for security versus our hunger for functionality (even for purely cosmetic features), and the stochastic nature of a resilient system does not bode well for its adoption. Quite possibly, it is this economic hurdle which is preventing progress, not a technical one.

Overall, there are certain properties that a system should possess in order to be considered resilient. It seems likely that redundancy is key—that is, that the failure of any discrete component should not cause systemic failure. In addition, the ways in which information is stored, accessed, modified, and transferred will all need to be carefully crafted so that a single failure or manipulation does not cause downstream consequences that are detrimental to the system as a whole or that allow for exploitation/modification of information. These methods are unlikely to look like traditional computer systems; instead, they are likely to appear less predictable at the component level, and have properties that are emergent rather than implicit. Data is unlikely to exist in just one spot, and different parts of the system will have to collaborate to decide what the ground truth actually is. Such work will be challenging, and lack of meaningful metrics will make comparison of approaches difficult. However, just because it is hard does not mean it is not worthwhile.

9. ACKNOWLEDGEMENTS

Matt Bishop was supported by grants from the National Science Foundation (CCF-0905503 and CNS-1049738), and Project Number 1776 of the GENI Projects Office. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of BBN Technologies, Corp., the GENI Projects Office, or the National Science Foundation. Marco Carvalho and Richard Ford were partially supported by the Department of Energy National Energy Technology Laboratory under Award Number(s) DE-OE0000511. This paper was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

10. REFERENCES

[1] E. Bonabeau, M. Dorigo, and G. Theraulaz. *Swarm Intelligence: From Natural to Artificial Systems.* Oxford, 1999.

[2] L. Boroditsky. How language shapes thought. *Scientific American*, pages 63–65, Feb. 2011.

[3] L. Briguglio, G. Cordina, N. Farrugia, and S. Vella. Economic vulnerability and resilience: Concepts and measurements. *Oxford Development Studies*, 37(3):229–247, 2009.

[4] M. Carvalho, T. Lamkin, and C. Perez. Organic resilience for tactical environments. In *5th International ICST Confernece on Bio-Inspired Models of Network, Information, and Computing Systems (Bionetics)*, Boston, MA, December 2010.

[5] L. Chen and A. Avizienis. N-version programming: A fault-tolerance approach to reliability of software operation. In *Digest FTCS-9*, June 1978.

[6] R. J. Ellison, D. A. Fisher, R. C. Linger, H. F. Lipson, T. Longstaff, and N. R. Mead. Survivable network systems: An emerging discipline. Technical Report CMU/SEI-97-TR-013, Carnegie Mellon University, 1997.

[7] F. Esponda, S. Forrest, and P. Helman. Negative representations of information. *Int. J. Inf. Secur.*, 8:331–345, September 2009.

[8] M. Franz. E unibus pluram: massive-scale software diversity as a defense mechanism. In *Proceedings of the 2010 workshop on New security paradigms*, NSPW '10, pages 7–16, New York, NY, USA, 2010. ACM.

[9] C. Gentry. Fully homomorphic encryption using ideal lattices. In *Proceedings of the 41st annual ACM symposium on Theory of computing*, STOC '09, pages 169–178, New York, NY, USA, 2009. ACM.

[10] A. Heddaya and A. Helal. Reliability, availability, dependability and performability: A user-centered view. Technical report, Boston University, Boston, MA, USA, 1997.

[11] E. Hollnagel, D. D. Woods, and N. Leveson. *Resilience Engineering: Concepts and Precepts.* Ashgate Publishing Group, 2006.

[12] A. R. Ives. Measuring resilience in stochastic systems. *Ecological Monographs*, 65(2):pp. 217–233, 1995.

[13] T. Jiang and J. Baras. Ant-based adaptive trust evidence distribution in manetpolicy-based mobile ad hoc network management. In *Distributed Computing Systems Workshops, 2004. Proceedings. 24th International Conference on s. Fifth IEEE International Workshop on*, pages 588–593, 2004.

[14] N. Khoussainova, M. Balazinska, and D. Suciu. Towards correcting input data errors probabilistically using integrity constraints. In *Proceedings of the 5th ACM international workshop on Data engineering for wireless and mobile access*, MobiDE '06, pages 43–50, New York, NY, USA, 2006. ACM.

[15] L. M. Mayron, G. S. Bahr, C. Balaban, M. Bell, R. Ford, K. L. Fox, R. R. Henning, and W. B. Smith. A hybrid cognitive-neurophysiological approach to resilient cyber security. In *The 2010 Military Communications Conference - Unclassified Program - Cyber Security and Network Management (MILCOM 2010-CSNM)*, pages 942–947, San Jose, California, USA, Oct.31st 2010.

[16] D. Mendonça. Measures of resilient performance. In *Resilience Engineering Perspectives: Remaining sensitive to the possibility of failure*, volume 1 of *Ashgate Studies in Resilience Engineering*, pages 29–48. Ashgate, 2008.

[17] V. Neagoe and M. Bishop. Inconsistency in deception for defense. In *Proceedings of the 2006 workshop on New security paradigms*, NSPW '06, pages 31–38, New York, NY, USA, 2007. ACM.

[18] B. Randell, P. A. Lee, and P. C. Treleaven. Reliability issues in computing system design. *ACM Computing*

Surveys, 10:123–165, 1978.

[19] P. Reitinger. Enabling distributed security in cyberspace. Technical report, Department of Homeland Security, 2010.

[20] C. G. Rieger, D. I. Gertman, and M. A. McQueen. Resilient control systems: next generation design research. In *HSI'09: Proceedings of the 2nd conference on Human System Interactions*, pages 629–633, Piscataway, NJ, USA, 2009. IEEE Press.

[21] C. E. Shannon. A mathematical theory of communication. *Bell Systems Technical Journal*, 27:379–423,623–656, 1948.

[22] J. A. Simpson and E. S. C. Weiner, editors. *The Oxford English Dictionary*. Clarendon Press, Oxford, UK, second edition, 1989.

[23] K. Tierney and M. Bruneau. Conceptualizing and measuring resilience – a key to disaster loss reduction. *TR News*, 250:14–17, 2007.

[24] K. S. Trivedi, D. S. Kim, and R. Ghosh. Resilience in computer systems and networks. In *Proceedings of the 2009 International Conference on Computer-Aided Design*, ICCAD '09, pages 74–77, New York, NY, USA, 2009. ACM.

[25] W. Weaver and C. E. Shannon. *The Mathematical Theory of Communication*. University of Illinois Press, 1st edition edition, 1949.

[26] D. Wei and K. Ji. Resilient industrial control system (rics): Concepts, formulation, metrics, and insights. In *Resilient Control Systems (ISRCS), 2010 3rd International Symposium on*, pages 15 –22, aug. 2010.

[27] B. L. Whorf. The Relation of Habitual Thought and Behavior to Language. In J. B. Carroll, editor, *Language, Thought, and Reality. Selected writings of Benjamin Lee Whorf*, pages 134–159. The M.I.T Press, Cambridge, M.A., 1956.

[28] D. D. Woods. *Creating Foresight: Lessons for Enhancing Resilience from Columbia*. Blackwell, 2005.