Antimalware Software: Do we Measure Resilience?

Richard Ford[1], Marco Carvalho[1], Liam M. Mayron[1] and Matt Bishop[2]
[1]Florida Institute of Technology
[2]University of California at Davis

About Authors

Richard Ford is the Director of the Harris Institute for Assured Information at Florida Institute of Technology, and the Harris Professor of Computer Science
Marco Carvalho is an Associate Professor in the Dept. of Computer Sciences, Florida Institute of Technology
Liam M. Mayron is an Assistant Professor in the Dept. of Computer Sciences, Florida Institute of Technology
Matt Bishop is a Professor in the Dept. of Computer Science, University of California at Davis
Contact Details: Dept. of Computer Sciences, Florida Institute of Technology, 150 W. University Blvd, Melbourne, FL 32901, USA, phone +1 321 674 8590, e-mail {rford, mcarvalho, lmayron} @fit.edu; Dept. of Computer Science, University of California at Davis, Davis, CA 95616-8562, USA, e-mail bishop@cs.ucdavis.edu

Keywords

Resilience, Metrics, Security measurement, Antimalware Software

Antimalware Software: Do We Measure Resilience?

Abstract

There is great interest in the topic of resilient cyber systems, especially with respect to attacks by malicious software. The challenges of measuring the actual resilience of a system and the ambiguity of the term "resilience" itself cloud much of the accompanying research. In this paper, we examine some of the lessons learned in defining resilience metrics. We argue that such metrics are highly contextual and that a general, quantitative set of metrics for resilience of cyber systems is impractical. Instead, a set of considerations and guidelines for building metrics that are helpful for a particular system are provided. We then consider these metrics in the light of current anti-malware software tests and argue that testing efforts have been primarily directed toward robust systems, not resilient ones. As such, our testing tends to push the market toward existing solutions geared toward prevention rather than mitigation and survivability.

Introduction

The design of complex computational systems has been experiencing a philosophical shift of principles, moving from a robustness-centred design to a more flexible and adaptive design. These systems are capable of surviving, reacting and recovering from external attacks and localized failures. This paradigm shift to a "fighting through design" philosophy is, in retrospect, unavoidable, as the limitations of proactive defence mechanisms become clear. However, one area that has not benefited from this mindset is the anti-malware space. When dealing with malware, once the defensive lines are breached, reinstallation or recovery from backups is the primary restorative action.

Given that it is now generally well-accepted that systems inevitably will be attacked, often successfully, it makes sense that they would be designed to survive these attacks, and recover from their effects to restore and maintain desired availability and functionality.

In this paper, however, we argue that this is not the case in the anti-malware world, and posit that while there is much good work in this area, real scientific progress has been hampered by the loose definition of "resilience" and the resulting lack of metrics, and consequently tests, in this space. As such systems become more accepted and deployed in different application domains, the need for a definition and metrics becomes of greatest importance, not only to establish common ground, but also to determine whether progress is occurring. This is reflected in the way in which we test anti-malware systems, which focuses almost exclusively on prevention, not cure (or co-existence).

A previous publication (Bishop et al., 2011) focused on the definition of the term "resilience" and how it relates to the concepts of "robustness" and "survivability". It noted that resilience is multi-faceted. Although often discussed form the perspective of performance and availability (Heddaya & Helal, 1997, Carvalho et al., 2010), resilience also relates to different properties of the system such as confidentiality and integrity (Bishop et al., 2011).

In this paper we focus on resilience metrics. After a brief review of the terminology and definitions, we introduce some of the current proposals for measuring resilience. We then discuss some of the challenges and limitations associated with these proposals, highlighting some of the additional considerations that must be taken into account to adequately represent the *resilience of a system*. Finally, we apply these insights to the state of anti-malware software, paying particular attention to the current state of the art in anti-malware tests.

What Is Resilience?

Resilience is challenging to define. The term may refer to specific systems, tasks, outputs, and other conditions that vary between scenarios, which precludes the development of a universal metric that applies to all system in all situations. Just as different musicians cannot agree on the "best" rendition of a song, this subjective definition of resilience has implications beyond mere style.

In some disciplines such as ecology, the resilience of a system is defined as the time the system takes to recover to steady state conditions after a perturbation. In computing, such a definition is unsatisfactory, partly due to the demands we place on our systems (the fitness of a system depends not on its endpoint, but on the path taken to get there) and partly due to the immaturity of computing recovery options. Biological ecosystems exist to reproduce—to continue to exist, essentially. Computing infrastructures typically have an external mission. In the cyber arena, if we were to define resilience to be just the recovery time, how would we factor in the differences in missions? This question, while difficult to answer, plays an important role in how we quantify and measure the resilience of a system.

So for our purposes, resilience is a measure of the ability to recover—but the specific measure, or measures, used to define resilience, and provide a value describing the resilience, must be determined from the particular system definition and the context in which the system is used.

Considerations for Resilience Metrics

The considerations of resilience metrics are particular to the context of cyber systems. Resilience of ecological systems, for example, rarely considers the magnitude of a response, focusing instead only on the time taken to return to pre-disturbance conditions. Developing a measure of such a property is not particularly difficult for these systems, and provides a way to compare and contrast different scenarios cleanly.

Before describing some measures for cyber systems, we point out that our notion of metrics is congruent with the extensive theory of measurement. As early as 1946, Stevens (Stevens, 1946) proposed different levels of measurement, ranging from nominal, the labelling of objects, to ratio, the use of more sophisticated statistical techniques to determine equality, rank order, equality of intervals and equality of ratios. Typically, we consider measurements to range from weak to strong, with the weakest being nominal, and progressing toward the strongest through ordinal, interval, and ratio.

We would like measurements of cyber resilience to be as useful as possible. When we refer to resilience, we need to ask what a system being "twice as resilient" as another actually indicates. If this cannot be expressed in terms that are meaningful, the idea of a ratio-based measurement may be impractical or inapplicable to the topic of resilience.

As a refresher, in measurement theory, a *measurement* is a representation of a quantity. It is *not* the quantity being measured, and this is an important distinction. A measurement provides insight into the attribute under inspection.

We propose several guidelines for constructing metrics that are appropriate for a particular system, given our definitional ambiguity. Each consideration is described and discussed.

Guideline A: All near-term metrics for resilience are likely to be ordinal

Engineers and scientists like to be able to assign numbers to things. "This GPU can carry out 1.1 teraflops—3 times as many as a CPU" reveals something concrete about the objects under comparison. It is 1-dimensional, because it compares only speed of computation. However, as we shall argue below, resilience is not a 1-dimensional quantity.

The resilience of a system would require "rolling up" a time series *f*(*t*) into a single number in order to be 1-dimensional. Different system inputs produce different time series; loss of dimensionality creates a many-to-one mapping and, consequently, a loss of information in the translation. Furthermore, the behaviour (and recovery) of the system varies depending on the failure or perturbation. For simple cases with a given set of possible outputs, it is typically possible to claim that one output is more desirable than another. This provides an ordinal metric.

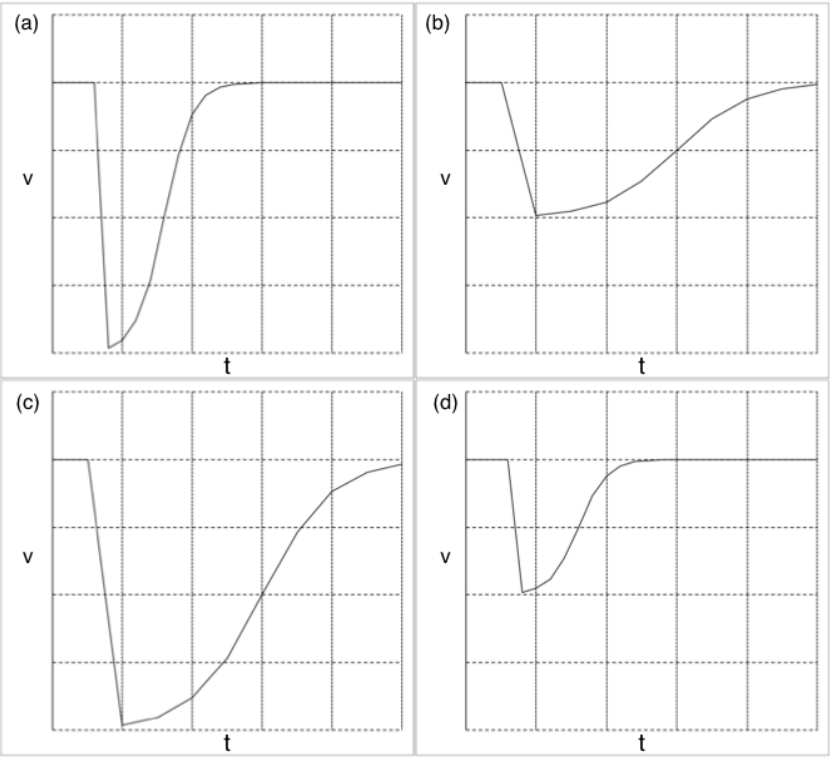

Figure 1: Examples or a system response to an external perturbation

Let us consider a very simple system as an example. We will use this example throughout the remainder of this paper to illustrate a straightforward system.

Consider a generator tasked to produce a certain voltage *v*. The system uses direct current (DC), so we do not have to consider issues related to phase and timing; instead, we have a single scalar value that represents the system at any particular time. At time t_1, the system undergoes a perturbation of arbitrary origin. Figure 1 shows four possible graphs for the system response.

In each panel of the graph, the basic shape is the same. However, in comparison to panels (a) and (d), panels (b) and (c) recover more slowly. Similarly, the peak perturbation in panels (a) and (c) is double that of panels (b) and (d). Finally, we note that these curves are idealizations – real systems may have unexpected and arbitrary responses to impulses, and any measurement scheme must take this into account. Measuring the lowest point and time to recovery does not adequately characterize these curves; neither does measuring the time to equilibrium. Even measuring the change in the area under the curve due to the perturbation, as proposed in (Wei & Ji, 2010), is only a partial measure of resilience (it is easy to think of curves which have the same area, but radically different properties).

Of course, instead of a single number, we can measure quantities *related* to resilience. Time to recovery, for example, is a numerical measure of a single *aspect* of resilience (Ives, 1995). Composing the measures of different aspects in order to reason about resilience itself is where our sense of ordinality originates.

Even with our simple example it is fairly easy to argue, by observation, that at least *in vacuo*, the system represented by Figure 1(d) is more resilient than those represented by Figures 1(a), 1(b) and 1(c). Recovery follows the same trend, but just happens more quickly or with less perturbation. Even here, though, things are not quite that simple, and we will revisit these graphs in Guideline H.

Guideline B: Resilience measurements are particular to a particular perturbation

Different perturbations cause different system responses. The failure and subsequent recovery of a particular part of a system's functionality is likely to lead to different output patterns. This inherent notion of events in resilience has been identified not only in the security domain but also in the areas of organizational (Westrum, 2006) and systems resilience (Sugden, 2001). In all cases, the concept relates to the challenge or disruption affecting the normal operation of the system.

Thus, when measuring resilience, we are actually measuring each individual perturbation and its different magnitudes, and providing an ordering that may be unique to a particular set of conditions. For example, one ecosystem might be resilient with respect to temperature increases of 5, 10, or 15 degrees Celsius, returning to steady state conditions after each temperature change. However, the same system may fail completely (that is, have no resilience whatsoever) in the event of a flood. For any system that we are likely to care about, there will be sufficient complexity that the resilience of the system will vary as a function of the type and magnitude of the perturbation. Determining the "best" system in this case will require an understanding of the disruptions that could occur in practice and of the users' tolerance of them. Capturing this numerically will be difficult.

Guideline C: Resilience metrics are deeply dependent on the boundary drawn around the system

Rarely does considering the resilience of a single piece of a larger system in isolation make sense. Our example above (see Figure 1) considered a generator in isolation. If we increase the scope of that system to include what the generator powers, our determination of its resilience with respect to the system changes. Consider, for example, a generator that is powering a series of incandescent bulbs. Such a system is still usable when the voltage sags—the lights may dim, but still provide adequate lighting. In contrast, a generator powering a computing device will fail in its mission when the voltage sags below a critical value—the computer is either working or it is not.

Let us extend our example by providing support for a battery backup. When the generator output sags, the batteries can provide power for a certain number of kWh. For this system, the total power shortfall drives the failure—that is, it does not matter if the generator output drops to zero as long as it returns to functionality before the batteries are exhausted. Continuing to expand our view, suppose the batteries can run long enough for a human to intervene and install a new generator. The electronic system is not in itself resilient—it does not repair itself or recover—but the system *as a whole*, including the human, is resilient.

The boundaries we draw around the “system” are critical in considering resilience. They must be carefully thought through and well defined. By changing the boundaries, we may change a system that we considered not to be resilient into one that may in fact be resilient, and *vice versa*. Any metrics we use to measure resilience are therefore specific to a particular system boundary.

Perhaps the most fundamental distinction we can draw concerns human input discriminating between those systems with autonomic recovery, and those requiring some level of manual intervention. Determining which type of system we are exploring is critical to our choice of metric. In the case of an autonomic recovery, we expect the system to handle perturbations without human input. By considering humans as part of the system, *all* systems are in some sense resilient because the imagination and understanding of people provide a deep pool of resources from which to rebuild the system. Conversely, for many real world systems, human intervention is a very real part of the larger system, and a resilience mechanism that provides adequate performance until humans can intervene is sufficient.

Guideline D: There is no universal way to combine multiple scenarios meaningfully to produce a “global” resilience ordering

As touched on above, any attempt to distill multiple measures of resilience into an “overall” measure of the system is fraught with problems. The varying magnitudes of each class of disruption may result in very different behaviors.

Attempting to unify the resulting curves into something that adequately represents the system is deeply contextual. Furthermore, when considering systems that need to be resilient to attack, any metric must take into consideration that a skilled attacker will attack the system at its weakest point. Attempting to reduce different aspects of resilience to a simple scalar loses so much information that we believe such a reduction to be ill-advised.

Guideline E: The ordinal ranking of a system could be different for each customer or application

System requirements drive our ordinal ranking of resilience, as they informally define our mental fitness function. Turning once again to our generator example, we can imagine two different sets of requirements. One customer may require the generator to maintain a certain minimum voltage at all times. Thus, any drop of voltage below this critical value makes the system as a whole not robust even if the generator itself recovers, and is not resilient if the system as a whole cannot recover from this disruption. In contrast, another customer may care about the total time the voltage sags below its assigned value. If this sag lasts longer than a certain period of time, the system fails. So in this case, even if the generator capacity recovers after the sag, the system as a whole has failed.

This leads to two observations. First, the systems are not the same. External dependencies beyond the generator itself make the systems different, even though the generation component is the same. Second, a particular behaviour of the generator can be “good” for some customers and “bad” for others. Thus, we cannot treat the customer requirements as a black box. The resilience of the generation system itself matters less than the resilience of the system it supports. The customer requirements must drive our metrics for resilience; they are not tied to a single component.

Guideline F: Metrics for cyber systems are different than those of their biological counterparts

When we consider cyber operations—especially when we must account for the presence of a malicious adversary intent on damaging the system—we must think about systemic resilience differently than when thinking about random failure.

When dealing with random errors, it is possible to determine with some degree of accuracy the probability of the different failure modes of the system. As such, one can consider the probability of different trajectories the system might take.

In contrast, when we apply this reasoning to *non-random* failures such as those that an attacker might cause, a different picture emerges. A simple example is helpful. Assume we have 10 vulnerable web servers, and we patch 9 of them. We might conclude that, because we have repaired 90% of the machines, our risk has diminished dramatically. Alas, when facing an adversary, this is incorrect. We have simply added a step that requires the attacker to identify the weak server. Then the site as a whole will be penetrated—so patching 9 web servers does not make the system as a whole 9 times more *secure*.

Thus, when dealing with an adversary, resilience needs to take into account attacker capability and cost.

Guideline G: Considering just the system output is not a sufficient picture of resilience

When reasoning about resilience, it is important to measure the ability of the system to withstand further attacks (Mendonca, 2008). For example, consider a system composed of *n* redundant generators. When a generator fails for any reason, it can be replaced by one of the other generators; conceivably, this could happen without any significant degradation of quality of service. However, after the failure, the system is not as resilient as it was previously because only *n*–1 redundant generators remain. This "capacity" of the system to recover from subsequent failures is an important part of determining systemic resilience and is not necessarily captured by the output of the system until it actually fails. As such, an important part of measuring the system's resilience is the cost in terms of its effect on the ability of the system to recover from subsequent failures or attacks.

Guideline H: Measuring resilience alone is usually not what we want

How we define "robustness" and "resilience" has strong implications when we try to compare the resilience of two systems. In particular, the sense that robustness is related to the system's rigidity, and the sense that resilience is related to recovery, can lead to some counter-intuitive conclusions if we attempt to measure resilience alone.

Consider the system producing the graphs in Figure 1. Imagine a system that is not affected (in terms of output) by any event or disturbance—that is, the system essentially continues unperturbed by the attack or failure. Technically, this system displays robustness, but has not demonstrated resilience to a particular attack. Thus, one could argue that it could be less resilient than a system that is perturbed by, but recovers from, the same kind of event. In this scenario, measuring resilience may not make sense, at least from the perspective of recovery to an attack (as defined in (Bishop et al., 2011)). An isolated measure of resilience may not be meaningful without a given context, and associated indicators of robustness of the system.

Anti-Malware Testing: A Case Study

Now that we have spent some time discussing resilience and its many dimensions, it is enlightening to apply these guidelines to a particular area. In our case, we turn our attention to anti-malware software, and ask questions not about the robustness it provides systemically, but about the resilience of the system.

Considering guideline C, it is important that we describe the bounds of the system and the property we are trying to provide continuity/recovery for.

The way we think about security is typically centred on robustness. Firewalls keep intruders out. Anti-malware software prevents hostile software from executing. Commercial products have historically focused on prevention. This actually is a reasonable perspective: stopping bad things from happening is crucial to security. However, post-incident behaviours are not as well understood but still important. When something goes wrong, how do we restore our system?

This question is complicated by the meaning of the word “restore”. While preserving and/or recovering services is valuable, in many cases it is best simply to restore infected machines from backups. If the data contained on the machine can be saved, application of a standard system image may well be the “safest” choice. Pragmatically, however, this is often not practical. Backups may not be available, and users often spend significant time customizing their machines for use. Thus, there is a time penalty to the “reinstall and reconfigure” approach. Conversely, administrators and executives care deeply about malware not being completely removed, which can allow subsequent attacks, data exfiltration, and even fiscal liabilities. As such, we want to measure not just the actual efficacy of system recovery, but the *confidence* that an organization can have in the repair.

By way of example, consider two products. One works 90% of the time, repairing the systems perfectly, but the remaining one time out of ten fails silently, allowing an attacker to retain access to the system. Conversely, another product can only recover 50% of the systems it is run on, but can detect perfectly whether or not recovery was successful. For most corporate environments, our sense is that the latter product is more useful than the former. However, even based on our guidelines, which is more resilient?

When we consider testing of individual products, it quickly becomes apparent that the community’s work to date has been focused on measuring the robustness of anti-malware products. We now look at four well-known and important reviewers and their tests, and consider their contribution to driving product resilience.

AV Comparatives

AV Comparatives (av-comparatives.org) is one of the major specialized anti-malware testing groups. AV Comparatives runs a large number of different tests of products; one of these tests is on virus removal. Unlike AV-Test, AV Comparatives infects a machine prior to the installation of anti-malware software (Anti-Virus Comparative, 2012).

Perhaps in part due to the difficulty of carrying out removal tests, only a handful of samples are used in disinfection tests, and all the malware used is relatively well known. Unlike detection tests that are run frequently, only two removal tests – from 2011 and 2009 – were available on the AV Comparatives website. Thus, while there is some limited focus on recovery, it falls a distant second to prevention.

AV-Test.org

One of the large specialist commercial anti-malware test groups is av-test.org. This effort, led by Andreas Marx, does measure some elements of resilience in that the tests include a section on repair. The tests are fairly simple, but do represent a major step in the right direction. According to AV-Test's descriptions, machines have anti-malware protection installed on them, and, then with the protection disabled, the system is infected. The products are then restarted and allowed to clean the machine. Finally, the cleaned machine is compared to the system prior to infection (AV Test, 2012).

While these tests do move us toward resilience, they also test a scenario that is rather unlikely to exist in the real world. Protection does not typically get turned off and on. Instead, a machine gets infected by a piece of new (to the protection subsystem) piece of malware that resides on the machine until the product detects it, either using known or generic malware detection techniques.

NSS Labs

NSS Labs carries out regular tests on anti-malware software. Many of their reports are "for fee", but one of the reports they frequently provide publically is their "Consumer Anti-malware Products Group Test Report" (NSSLabs, 2010). This report, typically available in the autumn of each year, documents the efficacy and performance of consumer-centric anti-malware protection. In the September 2010 tests, 11 well-known products were ranked based on their ability to block malware, block exploits, and their performance impact.

One of the interesting parts of the test is that it is exclusively measuring robustness. Should a machine become infected or a vulnerability exploited, the tests provide no information on how the products allow for system recovery. As such, this test clearly focuses on the robustness of protection, and not on its contribution toward resilience.

Virus Bulletin

Virus Bulletin has long been known for the high quality of its comparative reviews. However, Virus Bulletin tests measure impact on system performance and detection, not repair. Tests describe a component that measures how well products detect new samples. From the Virus Bulletin website (Virus Bulletin, 2012), we see:

> The VB RAP (Reactive And Proactive) test has been running as part of the bi-monthly VB100 comparative reviews since February 2009. The test measures products' detection rates over the freshest samples available at the time the products are submitted to the test, as well as samples not seen until after product databases are frozen, thus reflecting both the vendors' ability to handle the huge quantity of newly emerging malware and their accuracy in detecting previously unknown malware.

Such tests are interesting, but once again focus on the robustness of the products, not the resilience they provide.

Implications

The current state of the art of anti-malware product testing reflects our "walls and moats" mentality with respect to security: keep the bad things out. However, this mindset is anything but forward thinking and represents a battle that we have, as defenders, already lost. Especially in the malware space, we must accept that attacks will succeed and we must develop systems that continue to work even when penetrated.

The challenge is that we neither test the resilience of a particular machine or the entire system. At the individual machine level, tests of repair are hard to come by, and are overshadowed by tests of products' abilities to block threats before exploitation. At the systemic level, the picture is perhaps more complex. The diversity of protection techniques plus the unbounded human-driven recovery process does provide for some resilience, but this is not measured, or even really considered by current tests. Furthermore, as we discussed in our exploration of resilience more broadly, all systems that include human actors have some features that provide resilience. Autonomous resilience, however, is not being measured at all with respect to malware.

Unfortunately, this lack of measurement coupled with a heavy focus on robustness has driven the industry toward solutions that are brittle; when they fail, they fail completely. The process of working with infected machines is poorly explored, and vendors have little incentive to improve in this area. Test results drive user perception, which in turn drives product development cycles.

Future Work

One of the challenges of a metric that provides an ordinal scale is that we can say that, under a certain set of circumstances, System A is more resilient than System B, but we cannot say how much better it is. This is particularly important when considering the cost-benefit ratio of System A compared to that of System B. If, in practice, System A provides only a small improvement over System B, its value may not be much higher than System B's. Conversely, if "more resilient" means that A will survive and recover and B will not, the difference in value of System A over B is potentially very high.

Our sense is that there is no universal way of quantifying these differences. The "correct" approach is one that takes into account the relative costs and likelihood of failure. This is further complicated when systems need to be resilient to attack as well as random failure or perturbation. For such a system, an approach that provides generally good performance but fails utterly in one particular attack scenario should be weighed by its worst-case performance coupled with the cost to the attacker in terms of resources, sophistication, or exploitability. For example, if a system fails catastrophically when a certain number can be guessed but the chances of successful prediction are low—say 2^{64} to 1—the failure, despite its severity, might not be very important in practice.

Perhaps the solution is to identify a nuanced set of definitions for resilience that takes context and other external factors into account. Simplifying the definition to describe a single dimension of the property (for example, the system recovery time) may provide a single comfortable metric, but will certainly fail to grasp the full meaning of resilience. Instead, if resilience is considered as a multi-faceted property of the system, it may require a more complex description and a set of nuanced metrics, but will better represent the different system perspectives, and operational contexts.

The relative lack of test scenarios for resilience is an area that is ripe for exploration. As we have illustrated in the simple case study of anti-malware products, the major testing groups are hardly measuring resilience at all. Those tests, which do include a repair component, do not really shed much light on the resilience problem, and tend to consider scenarios that are, perhaps, unrealistic.

Given that the resilience of a system is so sensitive to the scenario under consideration, standardized scenarios being available for different problem spaces would allow the direct and *reproducible* comparison of different approaches to survivability (and robustness, and resilience) using different techniques. Without this, much of the work in resilient systems is open to criticism on the grounds that careful (or even random) selection of the scenario and requirements can lead to vastly different conclusions. Creating such a protocol for anti-malware, for example, would help drive user awareness, which in turn will shape the development of new product capabilities. Our ability to measure is as important as our ability to design and protect. Any funding agency interested in this space should carefully consider this point; even in our simple examples, two identical component behaviours can have different implications based on the design of the system as a whole.

Conclusion

In this paper, we have examined the concept of resilience as it applies to cyber systems. Our conclusion is that the development of an overarching set of metrics that can adequately measure resilience in all, or even most, systems is, for the foreseeable future, impractical. Resilience is very much about the requirements of the system, and different inputs can and will produce different systemic behaviour. Any “simple” measure of resilience obscures much detail, and is likely to be counterproductive.

In recognition of this, we have described several issues to be considered when attempting to measure the resilience of a simple system (a simplified power generator). We do not claim this group of issues to be universal or comprehensive, but it at least allows us to begin reasoning about both how to demonstrate resilience in experiments, and how to best compare different routes toward resilient systems. Despite the challenges inherent in measuring resilience, the problem is an important one, and its lack of a clear or partial solution restricts progress in the field of cyber resilience in general.

In particular, we have examined current tests of anti-malware product tests in the light of our resilience metrics guidelines. As can be seen, almost all our current testing is focused on measuring the robustness of products, not the resilience they can provide to the ecosystem in general. This testing bias needs to change if we are to provide evolutionary pressure toward solutions that focus more on survivability and resilience.

The complexity and nuances of resilience in real world systems are a major challenge in the development not only of resilient systems but also in allowing us to compare the actual behaviour of systems. This complexity rises quickly in the face of an adversary who will attempt to exploit the system in different ways. Our intuition tells us that funding agencies that have in interest in the design of resilient systems will need to provide unambiguous and shareable scenarios that allow the direct comparison of different techniques. These scenarios are a critical component of the definition of both “resilience” and of the actual system under consideration.

Acknowledgments

Dr. Richard Ford and Dr. Marco Carvalho were partially supported by the Department of Energy National Energy Technology Laboratory under Award Number(s) DE-OE0000511.

Dr. Matt Bishop was partially supported by National Science Foundation grants CCF-0905503 and CNS-1049738 and by the GENI Projects Office project 1776 under NSF Grant Number CNS-0940805 to the University of California at Davis. All opinions expressed are those of the authors and not necessarily those of the National Science Foundation.

References

AV Comparatives, (2011). Malware Removal Test, Autumn 2011. Downloaded from http://www.av-comparatives.org/comparativesreviews/removal-tests August 1st, 2012

AV Test (2012). Repair. Downloaded from http://www.av-test.org/en/tests/test-modules/repair/, August 1st, 2012

Bishop, M., Carvalho, M., Ford, R., & Mayron, L., (2011) Resilience is more than availability. In Proceedings of the New Security Paradigms Workshop (NSPW).

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

Heddaya, A. & Helal. A., (1997). Reliability, availability, dependability and performability: A user-centered view. Technical report, Boston University, Boston, MA, USA.

Ives. A. R., (1995). Measuring resilience in stochastic systems. Ecological Monographs, 65(2):pp. 217-233.

Mendonca, D., (2008). 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.

NSS Labs, (2010). Consumer Anti-Malware Products: Group Test Report Q3 2010, downloaded August 1st, 2012

Stevens, S. S., (1946). On the theory of scales of measurement. Science 103, 2684, 677–680

Sugden, A. M., (2001). Resistance and resilience, Science, vol. 293.

Tierney, K., & Bruneau, M., Conceptualizing and measuring resilience - a key to disaster loss reduction. TR News, 250:14-17, 2007

Virus Bulletin (2012), VB RAP test results. Downloaded from http://www.virusbtn.com/vb100/rap-index.xml August 1st, 2012.

Wei, D., and Ji, K., (2010). Resilient Industrial Control System (RICS): Concepts, formulation, metrics, and insights. In Resilient Control Systems (ISRCS), 2010 3rd International Symposium on, pp. 15 –22.

Westrum, R. (2006). A typology of resilience situations. In Hollnagel, E., Woods, D., Ashgate.