

# From Auto-adaptive to Survivable and Self-Regenerative Systems – Successes, Challenges, and Future

Michael Atighetchi

Information and Knowledge Technologies Business Unit  
BBN Technologies  
Cambridge, MA  
matighet@bbn.com

Partha Pal

Information and Knowledge Technologies Business Unit  
BBN Technologies  
Cambridge, MA  
ppal@bbn.com

**Abstract**—This paper charts the course of adaptive behavior in intrusion tolerance, starting from pre-programmed and user-controlled reactive adaptation to highly sophisticated autonomic and cognitively driven adaptation. The goal of intrusion-tolerance is to provide mission continuity even under conditions of sustained cyber attacks. We describe key themes of our previous work in adaptive cyber defense and introduction of autonomic response capabilities and discuss challenges that warrant further research. We also discuss the potential impact of new trends in distributed systems, e.g., service-oriented architecture and cloud computing, on future survivable systems, and point out new opportunities for developing sophisticated auto-adaptive capabilities for increased survivability .

**Keywords:** *Intrusion Tolerance, Information Assurance, Survivability, Cognitive Algorithms, Autonomous Computing*

## I. INTRODUCTION

Defending an information system against cyber-attacks is an arms race that is inherently asymmetric and favors the adversary. Approaches for mitigating this asymmetric threat have changed over the last two decades from a static model of security that tries to build failure free systems to a more dynamic and adaptive view of defense that assumes that failures are inevitable and the key for surviving attacks lies in the way failures and their consequences are handled. Recent improvements in technology areas such as service-oriented architecture, cloud computing, virtualization, and semantic linking have enabled more dynamic ways of system construction and resource management. Security and survivability architectures and associated designs and mechanisms need to keep up with the increased agility, dynamism, and fluidity of commoditized resources. We believe systems of the future can benefit by drawing upon the control and defense mechanisms that exist in nature and evolve over time. Cognitive approaches for cyber event interpretation and action selection, learning, and emergent properties all fall into scope as building blocks that, if assembled correctly, will lead to realizing the self-healing, self-managing, self-improving (collectively called self-\*) properties of survivable systems.

This paper summarizes main thrusts of our work in intrusion tolerance over the last decade and describes a steady

progression of increased agility, situational awareness, and autonomic defensive adaptation being built into information systems. Despite good progress, several hard challenges still remain relevant in today's environments and warrant further research. In addition, trends in recent distributed middleware technologies have made certain security concerns easier to address, e.g., via appropriate use of redundant resources, while raising the difficulty significantly in others areas, e.g., identity and privacy management and data confidentiality.

The organization of this paper is as follows: We present our work in section II, and describe hard challenges for providing cognitive defense components in section III. Section IV describes new trends in distributed systems and new opportunities for building survivable systems. Section V concludes this paper.

## II. PAST WORK IN INTRUSION TOLERANCE

This section describes our past work following the logical progression in the form of

- a) Creating a systematic approach for defense-enabling
- b) Creating reusable, autonomic dynamic responses
- c) Adding unpredictability to dynamic responses to make initial compromise more difficult
- d) Tolerating arbitrary corruption
- e) Integrating artifacts into a survivability architecture
- f) Automating cyber security decision making

### A. Defense Enabling

One of our earliest insights was to treat intrusion tolerance or survivability as an intrinsic property of applications and, in turn, of distributed systems that consist of interoperating applications. Consequently, we developed a technique to integrate security and survivability mechanisms with applications such that the applications can participate in their own defense. This software engineering process, called *defense-enabling*, has been the center-piece of our work in survivability and intrusion tolerance. Defense-enabling integrates defenses with functionality of the defended application for a given threat and resource environment. We use a middleware-based integration approach that facilitates

encapsulation of defenses into generic components that can be (re)used by multiple defense-enabled applications. The first step in defense enabling is to identify which potential attacks are most fatal for the application and rank them by likelihood and cost to the attacker. For example, attackers may be able to easily kill processes but it might be significantly more difficult to cause arbitrarily corrupt behavior. The next step is to develop defense strategies, focusing on the set of likely attacks that impact the system most first. The final step is to realize the strategies in middleware layer, which provides a clean separation between the application's functionality and the additional defensive capabilities. This separation allows independent development of defenses separate from application development, and facilitates reuse.

### B. Autonomic Dynamic Response

The experience of defense-enabling multiple applications led us to encapsulate a number of reusable defenses strategies and tactics in reusable packages [1]. *Mechanisms* provide individual capabilities provided by commonly available tools, such as firewalls or anti-virus software. *Tactics* are reactive defenses that use the capabilities of a small number of mechanisms. Typically, a tactic combines a sensor with an actuator mechanism to adapt to local situational changes. The basic objective of a defense *strategy* is to increase an application's survivability through automated coordinated defensive behavior. The overall strategy is to significantly improve the first line of defenses by managing the shortcomings and failures of the protection they provide. Our work supports hierarchical decomposition and realization of an overall defense strategy into a set of relevant sub-strategies, tactics, and mechanisms. This notion has similarities to orchestration and choreography in service-oriented architectures (SOAs), which we discuss in section IV.

### C. Use of Unpredictability

Defensive responses that are deterministic eventually become ineffective because, given sufficient time to observe the system and its defenses in action, the adversary will be able to predict defensive responses and plan accordingly. For instance, knowledge about where new replicas are spawned in a replication system enables attackers to plan and script complicated multi-step attack scenarios that maximize attack effectiveness at a low cost. In the ITUA project [2], we studied the use of unpredictability in adaptive responses at various OSI layers to increase the level of difficulty for creating scripted attacks. We injected uncertainty on the network layer by utilizing a NAT-based port and address hopping scheme that dynamically changed endpoint information at regular intervals and changing low-level socket behavior caused by deny rules in firewalls, e.g., ICMP reject vs. silent dropping of packets. At the application layer, we implemented replication managers that implement Byzantine fault-tolerance protocols to kill corrupted replicas and start new replicas following a probabilistic approach that imposes uncertainty on attackers.

### D. Byzantine Fault Tolerance

Straightforward use of redundancy can tolerate crash failures. But a sophisticated adversary can cause more damage

by corrupting the application that lead to Byzantine failures. To address this shortcoming, we developed a replication architecture under the ITUA project that can tolerate arbitrary (Byzantine) corruption of replicas. This framework transparently replicates existing application objects through gateways, actively manages replicas by killing corrupted replicas and starting new ones, and incorporates voting algorithms that can tolerate arbitrary corruption of replicas. Hosts are organized into security domains, which provide containment and diversity boundaries. Each host runs an ITUA Manager process, which uses the gateway to implement a specialized voting protocol for determining replica lifecycle actions. Replicas and application clients are connected via the ITUA gateway as well and implement the Byzantine fault-tolerant replication and connection groups.

### E. Survivability Architecture

After developing a number of defense mechanisms and using them in support of various defensive strategies and tactics, we attempted to see whether a survivable DoD-relevant information system can be developed, using currently available defense mechanisms and security technologies.

As part of the DARPA OASIS Dem/Val program and BBN's DPASA project [3], we designed and evaluated a high-watermark survivability architecture that provides strong guarantees for attack tolerance and survival, intrusion detection and situational awareness. The resulting architecture was based on the following survivability design principles:

- Single point of failure protection through increased redundancy for critical components
- Creation of physical barriers by creation of zones to make privilege escalation more difficult
- Controlled use of diversity to make each access path to key assets appear different
- Creation of a robust basis for defense-in-depth by implementing base algorithms as trusted hardware
- Enforcement of containment regions to limit spread of attacks across boundaries.
- Adaptive responses that quickly mount actions that have localized impact or can be easily reversed while deferring to humans for more coordinated and dangerous activities.
- Automated configuration generation from specs to limit inconsistencies

### F. Cognitive Approaches to Survivability Management

The state-of-the-art survivable system, built and evaluated in the OASIS Dem/Val program, showed excellent resiliency in containing and thwarting attacks, even when the adversary was given considerable access and system privileges. Expert human involvement was needed, however, to interpret alerts and incident reports and to make decisions about defensive responses. Human involvement in this manner is expensive, and motivates the essence of autonomic control envisioned in the next generation of self-regenerative survivable systems.

The goal of the CSISM project [4] was to develop automated reasoning mechanisms that when incorporated in the survivability architecture will minimize the role of human experts and pave the way for truly self-regenerative survivable systems. Our approach for taking system-wide actions was to map alerts and observations that come from the defense-

enabled system into a small number of abstract concepts, and structure reasoning around instances of these abstract concepts and their inter-relationships. This made the state space to be explored by the reasoning engines manageable, and also facilitated the use of the reasoning engine implementation and the underlying knowledge representation in other systems (because the implementation and knowledge representation are not tied to system specifics). CSISM implemented multi-layer reasoning, with fast reaction rules designed to take effective defensive actions within 250 ms of attack initiation, and a more deliberate cognitive reasoning process based on interpretation and hypothesis generation, response selection, and learning to take system wide defense actions.

### III. CHALLENGES

Our work in survivability and intrusion tolerance has demonstrated that it is possible to build autonomic intrusion tolerant systems that hold up to sophisticated and unrestricted attacks by dynamically responding to provide graceful degradation, failure isolation, and self-healing. However, in implementing example systems and subjecting them to attacks, we encountered a number of technical challenges for which solutions are necessary to advance the current state of intrusion tolerance to the next level of sophistication.

#### A. Challenges in using Cognitive Approaches

Learning in adversarial environments: To cope with unforeseen events, it is beneficial to include a learning component into the defense-enabling process to enable applications to learn from past mistakes as well as successes. In addition to standard problems of learning systems, such as the quality and quantity of the training set, learning in the cyber defense domain needs to account for the fact that intelligent adversaries are part of equation. Attackers may trick the learning system into learning the value of unimportant actions while holding back the real attack actions (and associated effects) to the very end. In CSISM, although the learner had access to all the information over all runs, it was unclear how much and especially which parts needed to be carried across various runs and whether it was safe for the learner to tweak the reasoning mechanism online—especially when there is no certainty about the occurrences or ordering of attack repetitions. Furthermore, learners rely on the notion of a quality signal, which indicates whether the system is offering good/acceptable service or not; and more importantly indicates to the learner when the service quality becomes unacceptable. Typical quality signals are multivariate, and incorporate a variety of views including system availability, costs to users, and damage. No such formalism exists in the context of cyber-security decision-making and in fact the notion of quality of cyber security or information assurance is highly subjective.

Simulation-based experimentation: To support validation of intrusion tolerance we created a high-level simulator that can model abstract concepts (e.g., hosts, networks, clients, and servers) and associated protocols between them (e.g., TCP flows, pub/sub interactions). We also developed a library of common attack effects that can be injected into the simulator, including host crashes, network partitions, file corruptions etc. The main idea was to use a simulator instance as an abstract executable model of the real environment. To support learning,

we added a snapshot capability and supported flexible quality functions over the system state. Given such a model, the online reasoning engine could perform what-if analysis by injecting an effect that corresponds to a recently detected attack and perform a pruned forward-chaining search to find the optimal sequence of defense actions that mitigates the attack. Besides the typical simulation fidelity issue, the biggest challenge we faced and did not successfully address was the automatic construction of the mapping between observable attack effects into injectable attack actions on the simulator.

#### B. Biologically-inspired selection functions

Encoding “self-preservative” invariants: Adaptive strategies can be used by intelligent adversaries and turned against the system they are trying to defend. We encountered one notable instance during the APOD red team exercises, where the red team tried to subvert the selection function for shutting down corrupted hosts and trick the system into self-inflicted denial of service. Introduction of stronger coordination in the formation of the selection function fixed this particular instance of the problem, but we expect similar problems to resurface in the future. We envision formulating and enforcing “do no harm” invariants analogous to safety invariants on mechanisms that mount defensive responses, especially the cognitive mechanisms responsible for cyber-defense decision making. No such framework exists in cyber defense, but biological organisms seem to exhibit this as an intrinsic trait.

Parameterized redundancy and diversity distributions: Redundancy is an important aspect in providing robustness and intrusion tolerance, but it comes at a high price. Diversity takes the same argument to an even more extreme level, despite the recent progress in artificially introduced diversity. A system that scales down its use of redundant and diverse resources in normal mode and scales up when under attack is clearly desirable from a cost conservation point of view. However, systems that rely heavily on such as strategy are vulnerable to stealthy attacks that only become visible when it is too late and no redundant resources are available any more. Although we had good success with 4-fold redundancy and graceful degradation in the DPASA survivability architecture, the initial cost of 4 times increase in hardware components have proven prohibitive in most environments. New research is needed to dynamically adjust redundancy and diversity distributions in a reliable and trustworthy way. Biological systems may provide inspiration in the form of dynamic management of diversity distributions of attributes (genes, behaviors) across populations.

### IV. NEW TRENDS AND OPPORTUNITIES

Recent advances in distributed computing have extended the Internet from a medium to connect multiple computers together to a global platform supporting flexible, interoperable, and dynamic interactions between clients and services. This new platform faces some of the challenges with respect to intrusion tolerance we previously outlined, but at the same time provides potential for making significant progress in some technical areas easier while exacerbating the need and difficulty in others.

SOA [5] and Cloud Computing [6] provide means to dynamically manage redundant computation resources in a

structured way. Cloud Computing can reduce a lot of provisioning issues and enable “on-click” dynamic provisioning of computing power and storage. The SOA concept implies that software building blocks, including security mechanisms, can now be thought of as services, potentially developed independently, to be connected to a service bus. Combining SOA and cloud has the potential to make intrusion tolerant architectures affordable, reusable, available, and dynamically adjustable to different environments and risk profiles.

However, indiscriminate migration to SOA and cloud computing can be dangerous. In addition to compute power, storage, or connectivity, the cloud must also offer a level of trust and protection. While distributing the processing of critical information across cloud nodes offers opportunity to load balance, improving availability, and to perform voting, improving integrity, it may significantly weaken confidentiality of mission critical data. Standard approaches for data confidentiality are based on cryptographic encryption (in transit or at rest) and delegation of service execution to innermost domains. Although encryption techniques remain applicable in the cloud, the data needs to get decrypted for a service to make use of it, and since the service is located somewhere in the cloud, this may open up the possibility of data leaks. New research in parallel algorithms and distributed split-data processing (analogous to split-key cryptography) could provide confidentiality guarantees because no single service in the cloud will have access to the all mission critical information units. This is clearly one area we are interested in pursuing. In addition, we see a strong need for advanced autonomic control algorithms that dynamically adjust redundancy and diversity distributions based on mission requirements and environmental parameters.

Moving to a SOA also means composing applications through a set of service interactions, which may break underlying assumptions made by individual services and introduce unforeseen vulnerabilities. Safe composition of an intrusion tolerant system of systems will require orchestration processes that preserve safety properties of components and flag inconsistencies in the composed systems. While languages like the Business Process Execution Language (BPEL) [7] offer ways to describe abstract executable business processes and workflows and their mapping to services, they currently only focus on representing publicly observable functional behavior of abstract processes in a standardized fashion and do not address systemic cross-cutting properties, such as robustness, security, or trust. This finding is similar to the argument made in [8] on the lack of resource management, exception handling, process variation, and data flow integration. Applying algorithms that operate on mission models and perform orchestration to maintain mission survivability is an exciting new research area that may change the way intrusion tolerant systems of systems are developed in the future. Biological systems may provide starting points for

construction of complex yet robust systems of systems, but careful analysis is key to avoiding the problem of metaphoric reuse without making improvements.

## V. CONCLUSION

In this paper we described our continued progression toward realizing the vision of better managed and agile distributed systems. Such systems can adapt their configurations, resource usage, even functional behavior to accommodate changes in their operating environments, including those that can be caused by a malicious adversary. As we begin to introduce new software development methodologies like cloud computing, on one hand there will be more opportunities to do better analysis of intrusion detection data, provisioning more compute or memory resources on demand leading to more intelligent system behavior and newer defensive responses. On the other hand, SOAs and cloud computing introduce new vulnerabilities around confidentiality, privacy, and trust. Biology may provide the starting point in terms of addressing such issues. Self-regenerative systems will need to be aware of and address the "auto-immune" issues where the system mistakenly (either by itself or by way of some fault) disables its defenses or attacks itself. The current software landscape and state of survivability research is at a point where investigation into these issues is not only possible but also becoming more important to undertake.

## REFERENCES

- [1] Michael Atighetchi, Partha Pal, Franklin Webber, Richard Schantz, Christopher Jones, Joseph Loyall. Adaptive Cyberdefense for Survival and Intrusion Tolerance. IEEE Internet Computing, Vol. 8, No. 6, November/December 2004, pp. 25-33.
- [2] Partha Pal, Paul Rubel, Michael Atighetchi, Franklin Webber, William H. Sanders, Mouna Seri, HariGovind Ramasamy, James Lyons, Tod Courtney, Adnan Agbaria, Michel Cukier, Jeanna Gossett, Idit Keidar, An architecture for adaptive intrusion-tolerant applications, Software: Practice and Experience, Volume 36, Issue 11-12 (September - October 2006) (p 1331-1354)
- [3] Jennifer Chong, Partha Pal, Michael Atighetchi, Paul Rubel, Franklin Webber. Survivability Architecture of a Mission Critical System: The DPASA Example. Proceedings of the 21st Annual Computer Security Applications Conference, Tucson, Arizona, December 5-9, 2005, pp. 495-504.
- [4] D. Paul Benjamin, Partha Pal, Franklin Webber, Paul Rubel, and Michael Atighetchi. Using A Cognitive Architecture to Automate Cyberdefense Reasoning. Proceedings of the 2008 ECSIS Symposium on Bio-inspired, Learning, and Intelligent Systems for Security (BLISS 2008), IEEE Computer Society, August 4-6, 2008, Edinburgh, Scotland
- [5] Erl, T. Service-oriented Architecture: Concepts, Technology, and Design. Upper Saddle River: Prentice Hall PTR. 2005.
- [6] Vaquero, L. M., Rodero-Merino, L., Caceres, J., and Lindner, M. A break in the clouds: towards a cloud definition. SIGCOMM Comput. Commun. Rev. 39, 1 (Dec. 2008), 50-55.
- [7] OASIS Standard WS-BPEL 2.0
- [8] Liming Zhu, Leon J. Osterweil, Mark Staples, Udo Kannengiesser, Challenges Observed in the Definition of Reference Business Processes, Workshop on Business Process Design 07, pages 95-107, 2008, Springer, Brisbane, Australia, September 24, 2007