Patterns of Self-Organizing Agile Security for Resilient Network Situational Awareness and Sensemaking

Rick Dove
Paradigm Shift International
Questa, New Mexico, USA
dove@parshift.com

***Abstract*—The security gap is widening as adversarial communities of all kinds employ more sophisticated techniques and broaden their targets of opportunity. These adversarial communities are structured as self organized proactive collaborators in tight learning loops driving rapid innovation – preying upon systems protected by wait and see strategies. In this paper we build upon six fundamental characteristics common to adversarial success, suggest that winning security can employ the same characteristics, show two patterns appropriate for resilient network support that fit this criteria, and describe a larger project that is developing a pattern language of next generation agile security. The two patterns described in this paper are modeled on the Biological Immune Systems and on mammalian hierarchical cortical sensemaking architecture. A technology that appears capable of implementing these patterns in relatively high biological fidelity is briefly introduced to support pattern-employment feasibility.**

Keywords- artificial immune system, feed forward hierarchy, hierarchical sensemaking, proactive anomaly search. Introduction

I. INTRODUCTION

Current system security strategies are falling behind, evidenced by the increasing costs spent on security and the increasing loses incurred by breeches. The reason for failure is evident: the attack community operates as an intelligent, multi-agent, self organizing, system-of-systems – with swarm intelligence, tight learning loops, fast evolution, and dedicated intent.

Here we address and define next generation security as co-evolving in this arms race with speed and innovation at least equal to the adversary, and accomplishing this with a fundamentally similar systems architecture.

Attack communities are diverse in nature and allegiance, but their strength is rooted in six common characteristics generally absent in current system security strategy: self organization, adaptable tactics, reactive resilience, evolvable strategies, proactive innovation, and harmonious operations. In a word, the adversary is agile.

How will system engineering facilitate sustainable system functionality in the face of intelligent determined attack?

In answer to these questions a research project began by establishing a framework for describing security patterns that exhibit similar characteristics, and initiated a search for candidate patterns that could serve as examples. That framework was populated by three initial examples [1], seeding subsequent work by the INCOSE (International Council on Systems Engineering) working group for Systems Security Engineering [2]. The project intent is to illuminate appropriate architectural and operational concepts that can be used as conceptual building blocks for next generation system security strategy.

The growing body of work in patterns for system architectures inspired a pattern descriptive approach arising from reviews of Christopher Alexander's seminal construction-architecture pattern work [3] and many others that have adapted his pattern concept to other fields.

Pattern work is generally a cataloging activity of repetitive proven expressions of architectural strategy, regardless of the domain. There is little in security best practice that exhibits the agile characteristics of the adversarial communities, so this project is an initial path finder activity rather than an organization of commonly respected and employed approaches.

However, natural systems have evolved successful security strategies over eons of contending with an uncertain and hostile world, and do offer proven strategies that can be examined for pattern potential. Natural systems are not limited to the biological systems that generally come to mind, but include social, economic, organizational, and political systems that have also evolved processes to maintain integrity in the face of attack.

The eventual purpose of the pattern project is the development of a pattern language that can be common and meaningful to both system engineers and security engineers, and comfortably informative to decision makers.

This paper introduces two new patterns to the pattern project, explaining that project's purpose and methods.

One pattern is inspired by recent understandings of attack and infection detection processes in the human immune system, with particular focus on the processes that identify new attacks and infections unseen previously. The other is modeled on the complex pattern recognition and sensemaking process now thought employed by the mammalian human neocortex, the current unmatched benchmark for making sense of noisy voluminous sensory input.

Both patterns are explained and displayed as abstract patterns for tailoring and reuse in appropriate applications.

Though these patterns can be implemented to greater or lesser degrees of biological fidelity, a new enabling technology is briefly introduced at the end of this paper that promises to bring relatively high fidelity within reach. This technology and these two patterns are relevant to resilient network needs, and form the core of an in-process resilient network application project (not yet published).

II. NEXT GENERATION CHARACTERISTICS

Current-generation security is characterized principally as reactive: it is invented and deployed in response to the escalating sophistication of attack experiences. As an after-the-fact defense insertion, it is typically an add-on functional subsystem, force-fit to the system that needs protection.

In contrast, next-generation security must at least provide parity with the agility of intelligent attackers and the communities that support their rapid innovation and evolution. The forefront of systems engineering is coming to grips with theory and abstractions for systems of systems and self organizing complex systems. The agility of the adversary and the urgency for effective systemic response offers the systems engineering community a tangible and urgent target for meaningful application.

The pattern framework outlined in this paper can provide a platform for system design requirements, while the pattern examples serve as demonstration.

The six characteristics exhibited by the adversary are translated into a mirrored agile security framework next.

A. *Self-Organization*

This is the most important of the six, and is a required system characteristic. It implies a dynamical system composed of components whose relationships reorganize in response to situational forces and events. Reorganization may be caused by willful decision-making agents embedded within the system, by systemic mechanisms that cause seemingly intelligent response, or by a combination of the two. Though decentralized control is favored for robust and innovative reorganizations, centralized reorganizing mechanisms can be effective if sufficiently rule-over-whimsy directed. Order within a self-organizing system is expected, on trend, to increase over time. In practice these systems are in a constant state of self-organization in response to opportunity and threat. If this activity ceases, the system is no longer agile. In a simple sense this causes adaptation. In a more important sense, this is the core of innovation evolution.

The Internet Storm Center [4] is a simple early example of a system composed mainly of independent agents, somewhat transient, both redundant and diverse in functional capability, that wax and wane in population according to the situation at hand. A different approach is the example proposed by [5] that employs component diversity in different versions of functionally equivalent software systems, which detect possible attacks when different outcomes occur, and identify the aberrant component; then a genetic algorithm replaces the offending component with a new variant devoid of the exploited vulnerability.

B. *Adaptable Tactics*

When an agile system is confronted with a novel situation it will reorganize its resources in a configuration appropriate to the situation. Adaptability is enabled by an inventory, or immediate acquisition, of appropriate resources. Typically adaptation is what occurs in tactical time frames, and is a real-time response to an opportunity or threat. When the situation allows time for a response, adaptation may include some modification of existing or available resources, provided that the new resource version is compatible with the overall system. Adaptation includes the use of existing available resources in new ways and for new ends.

C. *Reactive Resilience*

Agile systems live effectively in a world of risk, prepared to recover from disruptive incidents. The term resilience as a systems characteristic has origins in ecological systems, where fires, drought, hurricanes, construction runoff and other such insults disrupt a smoothly functioning ecological system. Ecological resilience allows the absorption of a shock that may alter the affected system for a while, but the system eventually returns to vibrant functionality. The phrase survivable systems is used in computer science, in general conformance with the concept of shock absorption and a possible period of performance degradation, but of course in a much faster time frame.

The immune system is a good role model of a self-organizing, resilient response process: it swings into action when an attack occurs and mounts an aggressive defense. A successful first-time defense learns from experience and is usually able to absorb subsequent attacks of the same nature with little or no performance degradation. Research in the new field of artificial immune systems is advancing quickly and has already resulted in new approaches to intrusion detection products [6].

D. *Evolvable Strategies*

Evolution takes time to develop new strategic avenues of capability, but don't think of it in biological time frames, slow by "nature." Think rather of John Boyd's OODA loop (observe-orient-decide-act) concept and the need to cycle evolutionary learning loops faster and tighter than the adversary does [7]. Boyd's OODA loop is typically thought of as a tactical concept, akin to competitive adaptability during adversarial engagement, but his fundamental model and the origins of his concept are based on cross-generation evolution of knowledge patterns [8].

Carl Woese [9] has suggested with some forceful arguments that biological evolution was most innovative and rapid in the period that preceded Darwinian evolution, when horizontal gene transfer (HGT) exchanged genes (as components) among single celled entities of different families. HGT works because genes are modular and interoperable with other genes in a cell (system); and because there is an exchange medium for transport of a gene from one system to another. Eventually components within a system become tighter coupled, more dependent on each other, and the system reaches what Woese calls the

Darwinian threshold. This is when the more familiar vertical evolution begins to dominate, and systems become architecturally complex with refinements on stable themes and less able to incorporate new innovations from the outside. We tend to equate evolution with the vertical Darwinian kind, and translate that into automated evolutions based on genetic algorithms. Woese has shown that horizontal evolution will reach an optimum in certain important characteristics and that vertical evolution cannot.

Two implications and one interesting conjecture fall out of this: (a) Adversary communities appear to be evolving toward and through a Darwinian tighter coupled refinement phase, establishing systemic vulnerabilities; (b) Next generation system security strategy might benefit by enabling and leveraging horizontal evolution in order to catch up, and must avoid being seduced by the sizable genetic engineering body of knowledge that practices Darwinian vertical evolution; (c) Adversarial evolution is based on recent ubiquitous connectivity and knowledge exchange offered by the Internet – if security strategy learns to take equal advantage, the asymmetry in current speed and innovation of attacker and attacked should disappear.

E. Proactive Innovation

The term proactive deserves careful attention, since it is often misused. People frequently misuse the term to simply mean active as opposed to passive. One way to know if someone is using the term proactive correctly is to ask whether the person is using it to mean an initiative (one that makes others become reactive) and/or an innovation (something both novel and valuable).

An excellent discourse on proactive security during aggressive engagement is John Boyd's classic eight-page essay "Destruction and Creation" [7].

F. Harmonious Operation

Embraceable, invisible, synergistic are words that come to mind for describing harmonious security. Usable security is the phrase generally attached to this concept, but sounds weak in comparison.

If a system's security mechanisms are not harmonious with the objectives of the people who use the system, they are not sustainable. Too much of the security effort these days is an imposition on user productivity. The effect is willful user rebellion, with too-frequent disregard and compromise of security policies, practices, and processes. If security compliance is tough and comes with a personal cost, willful compromise will occur, as well as unintended mistakes.

Natural systems have evolved examples of harmonious security: the immune system, for instance, doesn't field an infection-fighting population of antibodies until they are needed, relying instead on detection capability that can trigger the generation of infection fighters. Human designed systems have also addressed this need: for instance, making fire-retardant glass every bit as beautiful and transparent as regular glass encourages people to use the fire-retardant type when appropriate. Harmony is a common design principle in construction architecture but not in system architecture—a topic worth exploring at another time.

III. AGILE SECURITY PATTERN LANGUAGE

Shown in Table 1, these Six SAREPH characteristics are used as filters for selecting candidate agile security techniques. Though no research conclusions offer guidance yet for suggesting how many or what combinations might be minimally necessary, or even if these six are sufficient, the following guidelines are employed for nominating a candidate pattern:

- It must manifest both the self-organizing characteristic and the harmonious characteristic, in order to be sustainably agile.
- It must manifest either or both of the evolving and adaptive characteristics.
- It must manifest either or both proactive and reactive characteristics.

Table 2 shows the "form" employed in the project for displaying patterns. In mid 2010 the project conducted a Phase 1 workshop review of a number of initial pattern-capture attempts with a group of volunteer pattern developers. Those showing most promise are being developed under Phase 2 as a set of pattern-specific papers intended to lay groundwork for a Phase 3 attempt at shaping the beginnings of a multi-level pattern language. The purpose of the pattern language is to demystify the concepts of self-organizing systems-of-systems as a system security foundation, and to open working relationships between systems engineers, security engineers, and decision makers.

The pattern form shown in Table 2 is populated with pattern descriptions in Tables 3 and 4. These two patterns are discussed next.

TABLE I. PATTERN QUALIFICATION FILTERS

[S]	Self-organizing – with humans embedded in the loop, or with systemic mechanisms.
[A]	Adapting to unpredictable situations – with reconfigurable, readily employed resources.
[R]	Reactively resilient – able to continue, perhaps with reduced functionality, while recovering.
[E]	Evolving with a changing environment – driven by situation and fitness evaluation.
[P]	Proactively innovative – acting preemptively, perhaps unpredictably, to gain advantage.
[H]	Harmonious with system purpose – aiding rather than degrading system/user productivity.

TABLE II. **PATTERN DESCRIPTION FORMAT**

Name: Descriptive name for the pattern.
Context: Situation that the pattern applies to.
Problem: Description of the problem.
Forces: Tradeoffs, value contradictions, key dynamics of tension and balance, constraints.
Solution: Description of the solution.
Graphic: A depiction of response dynamics.
Examples: Referenced cases of pattern employment.
Agility: Qualifying SAREPH characteristics.
References: Links to source cases in the literature.

IV. PROACTIVE ANOMALY SEARCH

Table 3 is populated with the pattern form.

Biological immune systems (BIS) are highly effective at detecting and neutralizing attacks and infections by microorganisms. The highest evolved form in mammals sports remarkably adaptable processes for detecting and identifying new attacks and infections not encountered previously. This is accomplished by a process that continuously generates large, random, diverse quantities of speculative detectors (antibodies). Detectors are carried into service by two carrier types: one is targeted at attacks (in the blood) and the other is targeted at infections (in the cells). Before release, speculative detectors are first tested to make sure they will not respond to elements of "self", prohibiting the false positive alarm that would then trigger an undesirable immune response. Detectors that pass the self-tolerant test are released into time limited service – and eliminated if they fail to detect a foreign invader (marked by an antibody-matching antigen) by the end of their programmed life-cycle.

Speculative detectors are somewhat fuzzy in specificity. They will detect a range of similar antigens. When a speculative detector encounters an antigen it is immediately cloned in large quantity with structured mutations. These mutations compete for optimally specific fit. The fittest becomes a long-term memory detector, to ensure that identical invaders in the future are immediately detected before causing infection. Successful detectors also trigger the immune response, a remedial action to eliminate antigen-marked invaders in both blood and cells.

Stephanie Forrest, of the University of New Mexico and Santa Fe Institute, opened the Artificial Immune System (AIS) door for cyber security purposes with seminal modeling of both the biological immune system processes

TABLE III. **PATTERN EXAMPLE: PROACTIVE ANOMALY SEARCH**

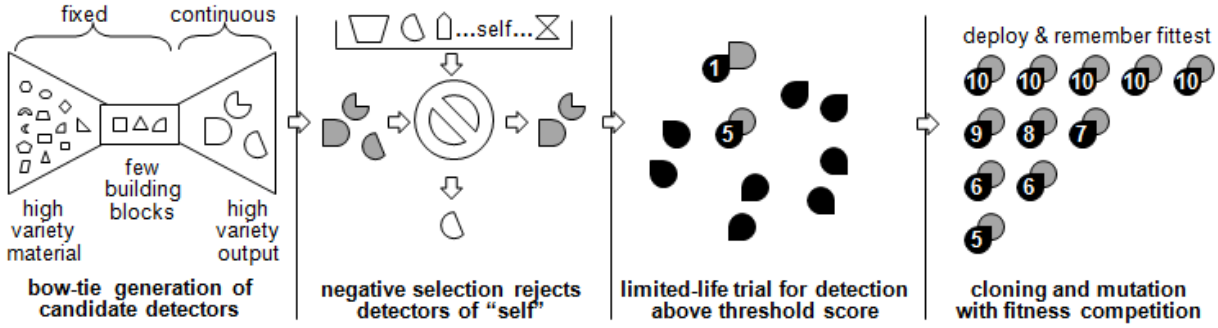
Name: Proactive Anomaly Search

Context: A complex system or system-of-systems subject to attack and infection, with low tolerance for attack success and no tolerance for catastrophic infection success; with resilient remedial action capability when infection is detected. Appropriate examples include cyber networks for military tactical operations, national critical infrastructure, and commercial economic competition.

Problem: Directed attack and infection types that constantly evolve in new innovative ways to circumvent in-place attack and infection detectors.

Forces: False positive tradeoffs with false negatives, system functionality vs. functionality impairing detection measures, detectors for anything possible vs. added costs of comprehensive detection, comprehensive detection of attack vs. cost of false detection of self.

Solution: A high fidelity model of biological immune system antibody (detection) processes that generate high quantity and variety of anticipatory speculative detectors in advance of attack and during infection, and evolve a growing memory of successful detectors specific to the nature of the system-of-interest.



Speculative generation and mutation of detectors recognizes new attacks like a biological immune system

Example: Lucid overview of antibody processes, including generation of speculative antibodies. See (Wikipedia 2010)
Example: Artificial immune system general model applicable to cyber networks. See (Hofmeyr 2000).
Example: Determining and evolving self and non-self behaviors in system call monitoring. See (Forrest 2008).
Example: Detector cloning and mutation improvement. See (Hightower 1996).

Agility: Self organization occurs in negative selection, in limited-life positive selection, in deployment cloning, and in memory of the fittest detectors. Adaptation occurs in bow-tie antibody (detector) creation, in negative selection and in positive selection. Reactive resilience occurs in constant refresh and replacement of useless and aged detectors. Evolution occurs as the memory of effective detectors grows with exposure to attacks and infections. Proactive innovation is the process of the bow-tie speculative antibody creation. Harmony is maintained by negative selection, and by limited-life purging of ineffective and of no-longer needed. [S-A-R-E-P-H]

References: (see reference section, only URLs shown here. All accessed 12Jun2010)
[19] (Forrest 2008) http://www.cs.unm.edu/~forrest/publications/acsac08.pdf.
[20] (Hightower 1996) http://cs.unm.edu/~forrest/publications/baldwin.pdf.
[21] (Hofmeyr 2000) http://cs.unm.edu/~forrest/publications/hofmeyr_forrest.pdf.
[22] (Wikipedia 2010) http://en.wikipedia.org/wiki/Antibody.

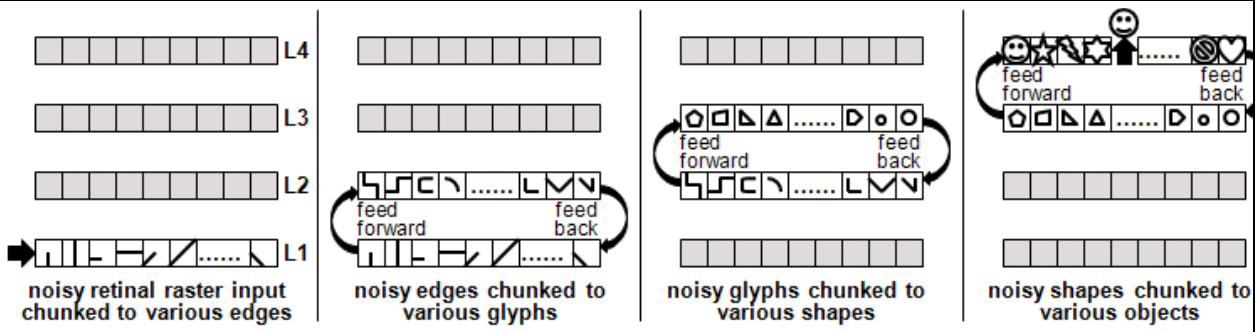
and translations into cyber-appropriate process models [10]. Her work, that of her colleagues, and those that have built upon that work, informs and guides the construction and display of this pattern.

The success of the biological immune system starts with its ability to generate and maintain a constantly refreshed large and diverse population of speculative detectors. Stable complex results are obtained from simple building blocks in a so-called bow-tie process [11], an agile security pattern in its own right [12]. A-C-T-G represent four organic molecules (like bytes) that can be sequenced into an infinite variety of molecular chains (like strings) of arbitrary length. This is the stuff of genetic code and of protein molecular combinations. The left side of the bow-tie (see graphic in Table 3 pattern form) represents all of the possible string combinations of A-C-T-G, while the bow-tie knot contains only a relatively small fixed subset of those infinite possibilities, separated into three different libraries whose contents (sequence segments) have evolved over eons into optimal modular building blocks for generating a high variety of antibody sequences. The right side of the bow-tie is where high variety detectors are generated dynamically from the building blocks of the knot. It appears that sufficiently-likely detector utility is obtained by a generation process that is not totally random, but rather structured to randomly select exactly one building block (string segment) from library A, one from library B and one from library C, combining them in that order into a speculative detector (antibody sequence).

V. HIERARCHICAL SENSEMAKING

The pattern form is populated in Table 4.

TABLE IV. **PATTERN EXAMPLE: HIERARCHICAL SENSEMAKING**

Name: Hierarchical Sensemaking
Context: A decision maker in need of accurate situational awareness in a critical dynamic environment.
Problem: A vary large amount of low-level noisy sensory data overwhelms attempts to examine and conclude what relevance may be present, most especially if time is important or if sensory data is dynamic.
Forces: amount of sensory data to be examined vs. time available to reach a conclusion, number of ways data can be combined vs. number of conclusions data can indicate, static sensory data vs. dynamic sensory data, noise tolerated in sensory data vs. cost of low noise sensory data.
Solution: Using a bow-tie process, each level looks for a specific finite set of data patterns among the infinite possibilities of its input combinations, aggregating its input data into specific chunks of information. These chunks are fed-forward to the next higher level, that treats them in turn as data, which is then further aggregated into higher forms of information chunks. Through feedback, a higher level may bias a lower level to favor certain chunks over others, predicting in essence what is expected now or next according to an emerging pattern at the higher level. Each level is only interested in a relatively small number of an infinite set of data-combination possibilities, but as aggregation proceeds through multiple levels, complex data abstractions and recognitions are enabled.
 Four level feed forward/backward sensemaking hierarchy modeled on visual cortex
Example: Cortical Spatial Sensing – Visual cortex receives noisy retinal raster of ~1,000,000 points and recognizes prior learned patterns in the field of view. See (Serre 2006). **Example:** Cortical Temporal Sensing – Cortex receives time sequenced sensory input and constantly predicts what is expected next according to prior learned patterns. See (George 2009). **Example:** Network Anomaly Sensing – Level 1 network agents detect anomalies on hosts, Level 2 agents interpret Level 1 alerts and cause inter-host collaboration, Level 3 agents set policy for Level 2 and interface with humans at Level 4, Level 4 is human decider on action and advisor to Level 3. (See Haack 2009).
Agility: Feed forward/backward interplay self organizes sensemaking resolution path through the four levels. Feed back adapts reactively to noisy input with suggested clean-up. (Learning evolves the content of levels, but is not part of this pattern). Proactive prediction of next temporal input feeds back expectations/suggestions to lower levels. Situational awareness and decision making level receives succinct and relevant information in harmony with processing capability. [S-A-R-P-H]
References: (see reference section, only URL shown here, all accessed 12Jun2010) [15] (George 2009) http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.163.7566&rep=rep1&type=pdf [17] (Haack 2009) www.cs.wfu.edu/~fulp/Papers/mims09f.pdf [13] (Serre 2006) http://cbcl.mit.edu/publications/ps/MIT-CSAIL-TR-2006-028.pdf

Extensive research into mammalian-brain pattern recognition and learning processes has matured to the point of generating useful cyber models. Notable is the visual cortex work of Pogio and Sere at MIT [13] demonstrating so-called "immediate recognition" of images with a four-level feed-forward pattern recognition hierarchy. In visual cortex, roughly speaking, the first level looks for specific learned types of edges in the retinal raster scan and passes these identifications on to level two, which looks for specific groupings of edges that are identified as learned glyphs of interest, which are passed to Level 3 that combines glyphs into learned shapes of interest, which are further discriminate at Level 4 as "immediate recognition" of learned objects of interest (e.g., car, face, cow). Additional levels come into subsequent play in determining more specific identity (brother Joe's face) and nuance (angry face) by rough example.

Feedback also occurs in biological cortex, from higher to lower levels when, for instance, noisy or ambiguous input prompts a higher level to "suggest" what to "favor" in pattern detection at a lower level. Deleep George, inspired by Jeff Hawkins [14, 18], did a seminal Stanford PhD thesis [15] with a more complete theory of cortical processing. Referred to as Hierarchical Temporal Memory (HTM), this cortical model includes both feedforward and feedback, and has purpose to learn and predict the patterns encountered in an uncertain world. Built as a working emulator at Numenta, this model is already employed in a third-party video-image-recognition product [16].

Of relevance, each level in cortical hierarchy deals with a single category of reconfigurable reusable features in manageable quantity, with intra-level collaboration among specific feature detectors. The architecture and process at all levels is identical, and employs a so-called bow-tie process [11], where a high variety of input is reduced to a small core set of modular building blocks, that can then be combined into a high variety of products – repeated at each level in the hierarchy.

A somewhat similar four-level model of this feed-forward/backward hierarchical architecture is used in [17] to offer succinct and highly relevant sensemaking information to human network administrators concerned with network attacks. Lower levels are composed of agents distributed at endpoints, with level-specific purpose leveraging partner-agent collaboration.

VI. Application Enabling Technology

The success of the biological immune system and of hierarchical temporal memory (HTM) as natural systems is dependent upon massively parallel *pattern matching* capabilities. Though computer processor chips continue to get faster each year, and chips with multiple processors deliver more parallel capabilities each year, their instruction sets and architectures remain far from suitable for necessary pattern matching needs. Attempts at fielding artificial immune systems (AIS) and HTM-like systems to date have necessarily made great compromises in biological fidelity imposed by current computational cost and performance barriers.

The human immune system is estimated to generate antibody receptors across a total "shape" space on the order of ten to the ninth different patterns. All of these are not present at any one time, but are rather generated over a cycle of replacement and regeneration on the order of 8 weeks or so, with perhaps ten to the seventh present at any one time. Affordable computation and memory for traditional pattern matching approaches has been elusive until now.

In June 2008 a General Purpose Set Theoretic Processor architecture for realization in VLSI was patented [23], featuring affordable massively parallel pattern detection in data streams. Projects funded by DHS developed FPGA prototypes, explored applications in security and related domains, and enabled successful licensing for production and marketing to a major VLSI producer.

Another project funded by DHS is currently (early 2011) investigating the feasibility of this new pattern processor to cover a much larger anomalous pattern space than even the human immune system addresses. For instance, detecting anomalous connection patterns in IPv6 packet headers using patterns composed of 32 bytes of address, 2 bytes for ports, and another byte for additional information creates a pattern space of ten to the seventy-seventh. Unpublished at this writing, the investigation has confirmed that this pattern space can be covered effectively by network endpoints sensing connections anomalous to their individual normal behaviors.

The pattern-matching processor architecture has the unique features of: (a) no-penalty constant data-stream throughput independent of the number and complexity of the simultaneous patterns being filtered in the data stream, and (b) affordable unbounded scalability with low-cost high-capacity pattern-detector chips that can be ganged in tandem should a single chip be insufficient [24].

Initial chip product is currently in advanced stages of production design, and promises to lift computational constraints on pattern-matching for both artificial immune systems and for hierarchical temporal memory approaches, as well as for many other constraint-bound pattern-matching applications.

Melanie Mitchell of Portland State University and Santa Fe Institute provides an appropriate synopsis in a recent Scientific American article [25]:

> "There's a huge amount of interest in new collaborations between biological and computer sciences because people are realizing computation goes beyond what we call 'computers'. One of the main things that all of these biological systems do so well is pattern recognition—pulling signal out of the noise even when they're inundated with information. Brains do it, individual cells do it, insect colonies do it—that's what *all* biological systems do in order to live. And we'd like computers to do that, too."

VII. Conclusions

This paper has suggested that six characteristics evident in adversarial communities are cornerstones of their success, and that mirroring these characteristics is a necessary

foundation for next generation security if parity or superiority is to be attained.

Based on self-organizing system concepts, these characteristics are uncomfortable to decision makers in their non-deterministic nature, and lack a shared understanding between systems engineers and security engineers. A method for developing a shared vision and common language among all three groups was described. Two example patterns modeled on natural systems were presented, demonstrating that self-organized non-deterministic systems can and do cope well with uncertainty and hostility.

Much work remains in pattern discovery, description, and organization before a comprehensive pattern language will emerge, but even early beginnings can provide useful building blocks that do not need to wait for the historical perspective before application.

As example, building on these two patterns plus the Bow Tie pattern [12] that was a sub-pattern to both, and a new pattern-matching processor about to enter the market, separate work is investigating application feasibility for immediate resilient network realization.

REFERENCES

[1] Rick Dove and Laura Shirey, "On discovery and display of agile security patterns," Conference on System Engineering Research. Hoboken, NJ, March 17-19, 2010. www.parshift.com/Files/PsiDocs/Pap100317Cser-OnDiscoveryAndDisplayOfAgileSecurityPatterns.pdf

[2] Rick Dove, "The buck stops here" Insight 13 (1), INCOSE, April 2010. www.parshift.com/Files/PsiDocs/Pap100101Insight-BuckStopsHere.pdf.

[3] Christopher Alexander. A Pattern Language: Towns, Buildings, Construction, Oxford University Press, 1977.

[4] Internet Storm Center. http://isc.sans.org/about.php

[5] Robert C. Armstrong and Jackson R. Mayo, "Leveraging complexity in software for cybersecurity," CSIIRW 2009, April 13-15, Oak Ridge, TN.

[6] S. Forrest, and S. Hofmeyr, S., "Engineering an immune system," Graft 4(5):5-9, 2001.

[7] J. Boyd, "Destruction and creation," Unpublished paper, 1976. www.scribd.com/doc/12627002/Destruction-and-Creation-by-John-Boyd.

[8] J. Boyd, J. "A discourse on winning and loosing," Unpublished paper. 1992. Abstract and first of five parts available at www.d-n-i.net/boyd/pdf/intro.pdf.

[9] Carl Woese, "Interpreting the universal phylogenetic tree", Proceedings National Acadamy of Sciences, 97(15):8392-6, 2000.

[10] S. Forrest, S. Hofmeyr, A. Somayaji, "Computer immunology," Comm. of the ACM 40(10): 88-96, 1997.

[11] Marie Csete and John Doyle, "Bow ties, metabolism and disease," TRENDS in Biotechnology, 22 (9), September 2004.

[12] Rick Dove, "Pattern qualifications and examples of next-generation agile system-security strategies," 44th Annual IEEE International Carnahan Conference on Security Technology, San Jose, California, USA, 5-8 October, 2010.

[13] Serre, T., Learning a Dictionary of Shape-Components in Visual Cortex: Comparison with Neurons, Humans and Machines, Ph. D Dissertation, Massachusetts Institute of Technology, June, 2006.

[14] Jeff Hawkins and Dileep George, "Hierarchical temporal memory – concepts, theory, and terminology," Numenta, 2006. www.numenta.com/Nument_HTM_**Concepts**.pdf.

[15] Deleep George, How the Brain Might Work: A Hierarchical and Temporal Model for Learning and Recognition, PhD thesis, Stanford University, 2009.

[16] Vitamin D, Inc., Unlocking the power of video with intelligent computing, Whitepaper from Vitamin D, Inc., available 12Jun2010 at www.vitamindinc.com/downloads/Vitamin%20D%20white%20paper.pdf .

[17] Jereme N. Haack, Glenn A. Fink, Wendy M. Maiden, David McKinnon, Errin W. Fulp, "Mixed-initiative cyber security: Putting humans in the right loop," Presented at International Joint Conference on Autonomous Agents and Multiagent Systems, 2009 (MIMS 2009). In press, 8th International Conference on Information Technology: New Generations (ITNG), April 11-13, Las Vegas, NV. http://u.cs.biu.ac.il/~sarned/MIMS_2009/papers/mims2009_Haack.pdf.

[18] Jeff Hawkins, Video talk: Hierarchical Temporal Memory and FDR (Fixed-sparsity Distributed Representations), University of British Columbia, Vancouver, March 18, 2010. Video: www.youtube.com/watch?v=TDzr0_fbnVk .

[19] S. Forrest, S. Hofmeyr, and A. Somayaji, "The evolution of system-call monitoring, "Proceedings of the 2008 Annual Computer Security Applications Conference, pp. 418-430, 2008.

[20] R. Hightower, S. Forrest, and A.S. Perelson, "The Baldwin effect in the immune system: Learning by somatic hypermutation," In Adaptive Individuals in Evolving Populations, R. K. Belew and M. Mitchell, (eds.), Addison-Wesley, Reading, MA, pp. 159-167, 1996. http://cs.unm.edu/~forrest/publications/baldwin.pdf.

[21] S. Hofmeyr and S. Forrest, "Architecture for an artificial immune system," Evolutionary Computation 7(1), Morgan-Kaufmann, San Francisco, CA, pp. 1289-1296, 2000. http://cs.unm.edu/~forrest/publications/hofmeyr_forrest.pdf

[22] Wikipedia, "Antibody," accessed 14Jun2010. http://en.wikipedia.org/wiki/Antibody.

[23] Curtis L. Harris and Jack Ring, General Purpose Set-Theoretic Processor, U.S. Patent 7,392,229, issued June 24, 2008.

[24] Rick Dove (2009), "Pattern recognition without tradeoffs: Low-cost scalable accuracy independent of speed," Cybersecurity Applications and Technologies Conference for Homeland Security (CATCH), Washington D.C., March 3-4, www.kennentech.com/Pubs/2009-PatternRecognitionWithoutTradeoffs.pdf.

[25] John Pavlus, "Borrowing nature's code, Scientific American, Ten World Changing Ideas, December 2010, p 49.