Self-Organizing Resilient Network Sensing (SornS) with Very Large Scale Anomaly Detection

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

***Abstract*—Anomaly detection promises to find elements of abnormality in a field of data. Computational barriers constrain anomaly detection to sparse subsets of total anomaly space. Barriers manifest in three ways – conserving both pattern memory capacity and pattern matching cycle time, while closing off scalability. The research reported here has discovered and analyzed a technology to eliminate two of these barriers, memory capacity and cycle time, and by targeting implementation at a new VLSI pattern processor, eliminate the third scalability barrier. An example shows how 10 to the 15 patterns integrated as a single gang detector can be stored in 193 bytes of memory, with much larger pattern magnitudes practical as well. The architecture of the gang detector enables complete processing of all 10 to the 15 patterns in time determined by the number of features in a single pattern, rather than the total number of patterns. Scalability is provided by a reconfigurable massively parallel VLSI pattern-matching processor chip that can accommodate a virtually unbounded number of such gang detectors. Anomalous behavior detection promises a way round the limitations of looking only for known attack patterns, but it raises new issues in the cyber domain of higher false positive rates and questionable normal-behavior stability. Work reported in this paper describes the nature and capability of gang detector employment, and suggests that the traditional issues of anomaly detection can be addressed with an architecture that engages in continuous learning and re-profiling of normal behavior, and employs a sensemaking hierarchy to reduce false positives. The architecture is based on process patterns from the biological immune system combined with process patterns of mammalian cortical hierarchical sensemaking.**

Keywords-anomalies; anomaly detection; artificial immune system; cortex; cortical hierarchy, packets; patterns; zero-day attacks;

I. INTRODUCTION

Anomaly detection promises to find elements of abnormality in a field of data. High end applications include finding patterns in unstructured data, identifying emergent behaviors in multi-agent systems, illuminating insider threats in progress, and detecting cyber attack vectors unseen previously.

Computational barriers constrain anomaly detection to sparse subsets of total anomaly space. Barriers manifest in three ways – conserving both pattern memory capacity and pattern matching cycle time, while limiting scalability.

The research reported here has discovered and analyzed a patent-pending technology to eliminate two of these barriers, memory capacity and cycle time, and by targeting implementation at a new VLSI pattern processor, eliminate the third barrier of scalability. An example discussed shows how 10^{15} patterns integrated as a single *gang detector* can be stored in 193 bytes of memory, with much larger pattern magnitudes practical as well. The architecture of the gang detector enables complete processing of all 10^{15} patterns in time determined by the number of features in a single pattern, rather than the total number of patterns. Thus, 10^{15} patterns that each have seven features require only seven simple byte-matching computational cycles. Scalability is provided by a reconfigurable massively parallel VLSI pattern-matching processor chip that can accommodate a virtually unbounded number of such gang detectors. Unbounded in that multiple pattern processors can be employed should one be insufficient.

Feasibility of gang detectors filtering network traffic was demonstrated with simulators during phase 1 of a DHS SBIR contract. The pattern processor chip is in final stages of production preparation, having demonstrated its capabilities in a prior DHS S&T SBIR contract. Combining gang detectors with the pattern processor makes practical previously unattainable anomaly detection in many domains.

Anomalous behavior detection promises a way around the limitations of looking only for known attack patterns, but it raises traditional issues of higher false positive rates and questionable normal-behavior stability. Mitigation of these two issues is not a topic covered in this article, but will have some discussion in the conclusion section where next phase project work is briefly outlined.

Phase 1 work reported in this article describes the nature and capability of gang detector employment, and suggests that the traditional issues of anomaly detection can be addressed with an architecture that engages in continuous learning and re-profiling of normal behavior, and employs a sensemaking hierarchy to reduce false positives. Notably, the architecture is based on process patterns from the biological immune system combined with process patterns of mammalian cortical hierarchical sensemaking [3].

This work was funded by the Department of Homeland Security under contract D10PC20039. The content of the material contained herein does not necessarily reflect the position or policy of the Government, and no official endorsement is implied.

This project was undertaken to generate innovative resilient network technology. Its focus on sensing and sensemaking recognized the imbalance between remediation options, such as those now promised by cloud computing and virtualization – and the ability to better sense when remediation is required.

II. HARDWARE FOR MASSIVE PATTERN DETECTION

This research project leverages a new VLSI pattern processing technology, referred to here as the pattern processor (PatProc), which promises to overcome the barriers to a high fidelity immune-system-like network anomaly detection capability. This technology enables the capabilities developed under the research project, and some background discussion is necessary in order to understand the research approach and results.

In June 2008 a General Purpose Set Theoretic Processor architecture for realization in VLSI was patented [6], featuring affordable massively parallel pattern detection in data streams. Phase 1 and phase 2 SBIR contracts (NBCHC070016) awarded 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. Initial chips are currently in advanced stages of production design, with availability sufficient for phase 2 prototyping.

The PatProc architecture [1] represented conceptually in Fig. 1 has the unique features of: (a) constant data-stream throughput independent of the number and complexity of the patterns being filtered simultaneously in the data stream, and (b) affordable unbounded scalability with low-cost high-capacity pattern-detector chips that can be combined in tandem should a single chip be insufficient.

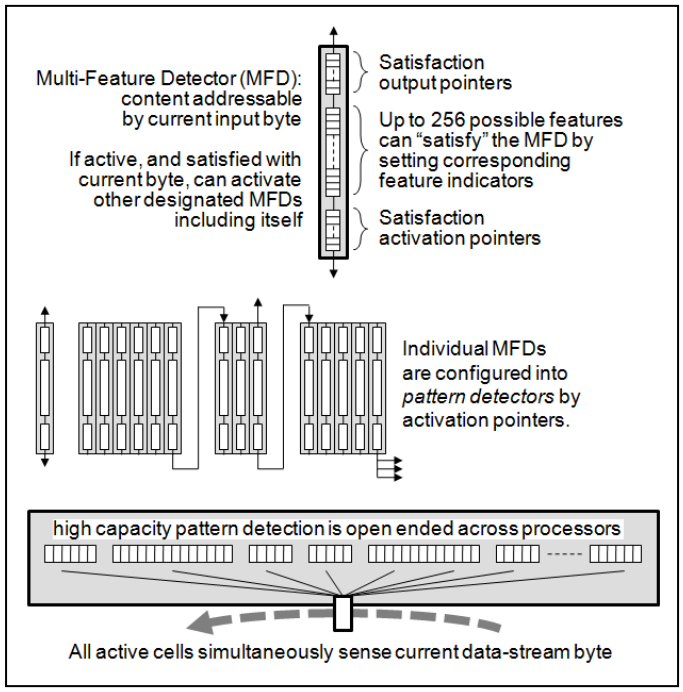

Figure 1. Reconfigurable pattern processor – reusable cells reconfigurable in an open-ended architecture

A single PatProc will accommodate a large number (potentially 10 thousand or more in first silicon) of "individual" anomaly detectors. As will be seen later, this project has found a now patent pending way to increase that number to vast quantities well in excess of the 10^{15} pattern-capacity examples that will be shown.

III. SPECULATIVE ANOMALY DETECTION

Speculative anomaly detectors, like biological immune system (BIS) antibodies, are randomly generated patterns that are generally not known from prior learned experience or from external sources of known intrusion patterns. The process for generating and managing the life cycle of speculative detectors developed in this study is inspired by the adaptive biological immune system. The concept is important for our work, as a prime objective is to detect never before seen attacks.

Biological immune systems 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 foreign-body invasions not encountered previously. This is accomplished by a process which inspired a modified version shown in Fig. 2 that continuously generates random, diverse, and large quantities of speculative detectors (antibodies). Before release into the blood stream, new detectors are first tested to make sure that they will not respond to elements of "self", prohibiting a 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 (antigen) by the end of their programmed life-cycle. This self-tolerization process is referred to as negative selection, as the detectors that remain have been selected by virtue of not reacting to self. They are then put into service looking for anomalies that don't belong in blood.

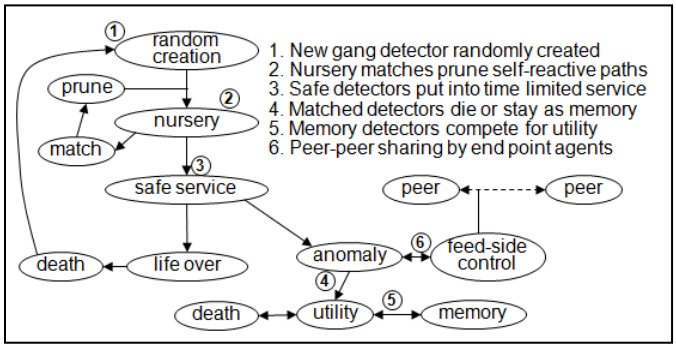

Figure 2. General detector life cycle

Stephanie Forrest, of the University of New Mexico and Santa Fe Institute, opened the artificial immune system (AIS) door for cyber security purposes in the early '90s, with seminal modeling of biological immune system processes and translations into cyber-appropriate process models [5]. Forrest's work and that of her colleagues, and those that have built upon that work, have informed and guided the phase 1 work. The Ph.D. dissertation of Steven Hofmeyr [7], one of Forrest's students, was especially informative. Hofmeyr

graciously participated as a subject-matter-expert in reviews of our work in process, offering helpful suggestions and encouragement, though no endorsement is implied.

Somewhat similar to BIS detectors (antibodies) ,which start out as fuzzy detectors that will match a range of similar antigens, we employ a gang detector (GD) that is an integrated collection of many patterns into a single detector. A GD is therefore fuzzy in that it is not clear upon detection which pattern was matched. However, unlike the BIS, we can immediately determine the exact individual pattern that caused the match by simply looking at what is in the data stream that triggered the gang detector's match event.

IV. GANG DETECTORS

Fig. 3 depicts a typical way the PatProc might be configured to detect a specific single pattern. The PatProc could be configured with many such single-pattern detectors, each sensitive to different patterns and all active simultaneously.

In Fig. 3 the depicted pattern detector consists of 7 multi-feature detectors (MFDs) connected by sequential activation links, each with a single feature indicator set. Collectively these 7 MFDs will detect a pattern (path through the MFD) of 7 contiguous feature-stream byte values that correspond to the feature indicators set in each MFD respectively. Fig. 3 depicts a single pattern detector that would be appropriate for detecting specific packet header information that has been extracted from the raw packet by a preprocessor (PreProc) and then fed to the pattern processor as 7 contiguous features (bytes). Many such single-pattern detectors could be present and active in the pattern processor simultaneously, all examining the same feature stream for different patterns.

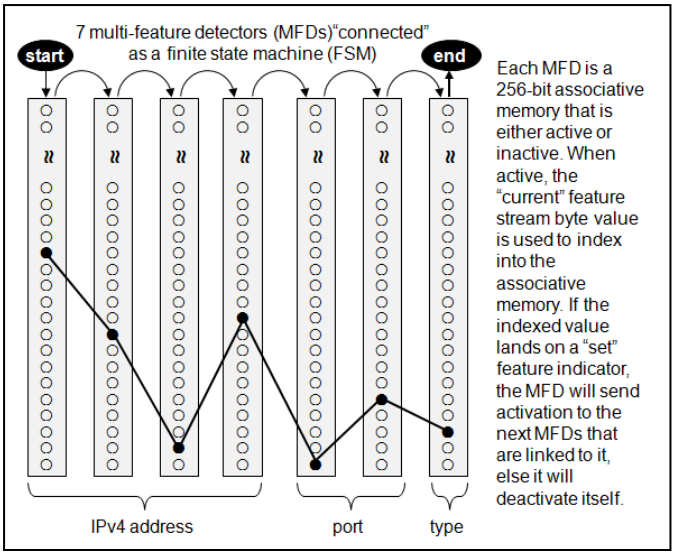

Figure 3. Single-pattern detector for a specific packet header connection

Multi-feature detectors are called such because they may have multiple feature indicators set, and thus are sensitive to multiple feature values rather than just one. A gang detector employs this multiple setting capability to detect many patterns integrated as a group within a single collection of MFDs. Fig. 4 depicts the concept and points out the unique benefits of this approach unattainable previously.

The vast numbers of anomaly detectors possible in a GD enable a high fidelity artificial immune system, something not witnessed with current conventional technology. All pattern space not associated with normal operational behavior patterns (self) must be covered completely by detectors looking for non-self patterns. If this can be accomplished, false negative anomalies would be eliminated, assuming all of the normal behavior patterns associated with self can be removed from the mechanism employed to detect anomalies. Unfortunately, as has been noted repeatedly in the literature, cyber network environments are not stable and normal behavior changes, unlike the biological immune systems.

Two issues must be addressed: complete coverage of anomalous pattern space, and a continuous reevaluation of normal operating behavior and its complimentary characterization of anomalous behavior patterns.

In a simple example, a gang detector might be created at birth with all feature indicators set, allowing it to cover 100% of all pattern space, both normal and non-normal (anomalous). The gang detector could then be exposed to normal behavior for some period of time, and every pattern that was detected during this self-tolerization, or training, period, would then be removed from the gang detector. At some appropriate time when patterns are no longer detected it might be assumed that the gang detector no longer contains any patterns associated with normal behavior and then be put into service.

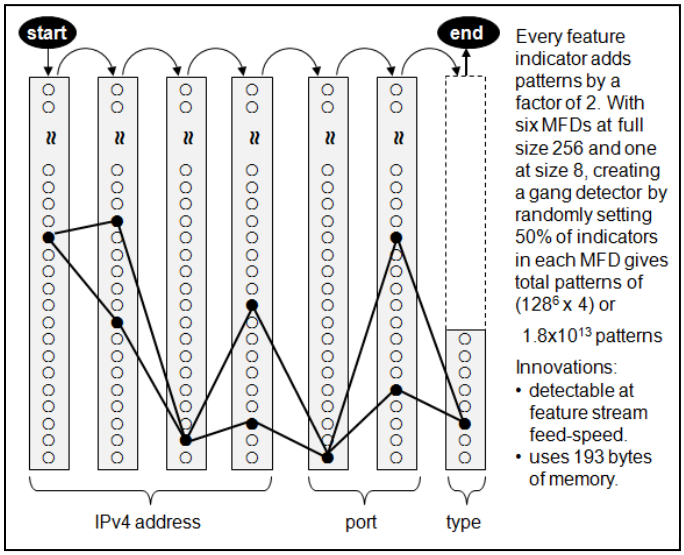

Figure 4. Conceptual depiction of a gang detector

A gang detector like that in Fig. 4, fully populated with feature indicators, covers 256^6 x 8 = $2.25x10^{15}$ unique Pattern Paths. … represented in just (6x32)+1 = 193 bytes. If each of these patterns were in a conventional pattern list, seven times the number of possible patterns in feature-byte storage would be required, or approximately: 10^{16} bytes in contrast. Computational time aside, these storage comparisons show the prohibitive barrier for achieving high-fidelity immune-system performance with traditional computational approaches

Assuming for the moment a stable environment that has presented all normal behavior, there is still a problem: removing a single specific pattern from a gang detector is not possible. It should be clear that removing a pattern does not require the removal of the entire path (all seven feature indicators). The removal of any single feature indictor in the path disables that pattern from further detection. However, removing a single feature indicator disables many more patterns, ones that may be legitimate anomalous patterns that use that feature indicator in combination with other feature indicators unrelated to the normal pattern that was disabled.

The solution to the removal problem is central to the issue of full anomalous pattern space coverage, and is statistical in nature. In short, full coverage is accomplished by using many gang detectors that each covers overlapping but different portions of anomalous pattern space. Fig. 5, borrowed from the artificial immune system literature, depicts a two dimensional representation of pattern space, and shows a portion reserved for self and the remaining portion partially covered by many other detectors likened to fuzzy antigen detectors, antibodies that will react to many different antigens before they are cloned and improved for a precise match through a process called somatic hypermutation [7:10].

BIS solves full coverage with two mechanisms: random generation of many antibodies with diverse but somewhat overlapping coverage, and continuous cycling over time through the entire pattern space with new fuzzy antibodies replacing those that have failed to encounter an antigen. Our approach is similar in achieving full coverage through statistical means, in that we generate many diverse gang detectors, but we are more efficient, in that we can field a sufficient set of gang detectors to statistically cover virtually all of the anomalous pattern space at any one time, should that prove to be necessary or useful.

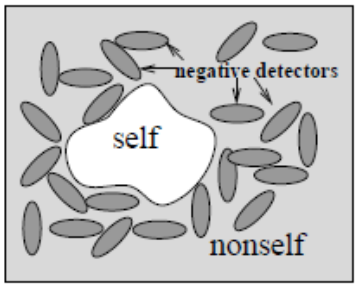

Figure 5. Pattern space and self-nonself discrimination, graphic from [4]

Fig. 5 characterizes BIS pattern space as a universe of data points partitioned into two sets – self and non-self. Negative detectors, those selected after tolerization for non reaction to self, individually cover somewhat overlapping subsets of non-self.

Subsets are continuously replaced over time with new subsets that cover different portions of non-self pattern space, eventually cycling through entire coverage of non-self space. BIS appears to cover a pattern space of something like 10 to the 9 patterns [2] over a period of 8-10 weeks or so.

Now consider a speculative GD creation process that randomly turns on 50% of the feature indicators within a 256-bit multi-feature detector, and repeats that process for all 7 cells in a 7-cell detector. If all 256 values in each of the first six detector cells could potentially be encountered, and 8 values within the seventh, such a detector would match 128^6 x 4 different patterns on average – considering that 50% allowable feature indicators on average are set in each multi-feature detector. That means one detector can match 1.76×10^{13} different patterns. There are 256^6 x 8 possible patterns in all of pattern space, which is 2.25×10^{15} patterns. Thus, a single 7-cell detector generated in this manner can cover 0.78% of all pattern space.

We investigated the feasibility of pattern-space coverage for some appropriate set of security signature domains. For matters of convenience, we chose to employ detectors for the IPv4 packet connection domain, with an eye on IPv6 scaling issues. The principal work was to understand how GD construction and pattern space behave interactively, and to determine appropriate bounds on GD construction for the intended first generation PatProc technology.

Three key questions to answer about GDs employed for anomalous behavior detection are associated with pattern space coverage:

- How is pattern space coverage affected by the number of multi-feature detectors in a GD? (two sizes are examined: 6 and 32)
- How many GDs are required to cover pattern space sufficiently? (see Fig.6)
- Can this number of GDs be practically accommodated by the pattern processor technology? (yes - easily)

A. IPv4 Gang Detector Coverage

Pattern space for a 7-feature pattern, where six feature may have 256 different values and one may have 8 values, encompasses a total pattern space of 2.25×10^{15} unique patterns. A statistical model was built in Excel to determine how much of pattern space is covered by different numbers of such 7-feature GDs from what is believed to be an optimal (for coverage) set of construction parameters, 50% cardinality, meaning half the feature indicators are turned on. The results of that model are shown in Fig. 6 at the top, and indicate that 512 GDs appear to provide more than sufficient coverage for the chosen construction parameters.

It should be noted that it is not necessary to provide full coverage with 512 GDs at any one end point at any one time if similar endpoints share the discovery of anomalies, so that 10 sharing endpoints would only require $1/10^{th}$ as many GDs to have the same coverage effect. Also each endpoint will cycle its in-service GDs over time to address the possibility of changes in the operational environment, repopulating the set of GDS in service. Thus the same effect of coverage is obtained at any one endpoint by ten cycles of 50 GDs as is obtained with a single cycle of 500 GDs

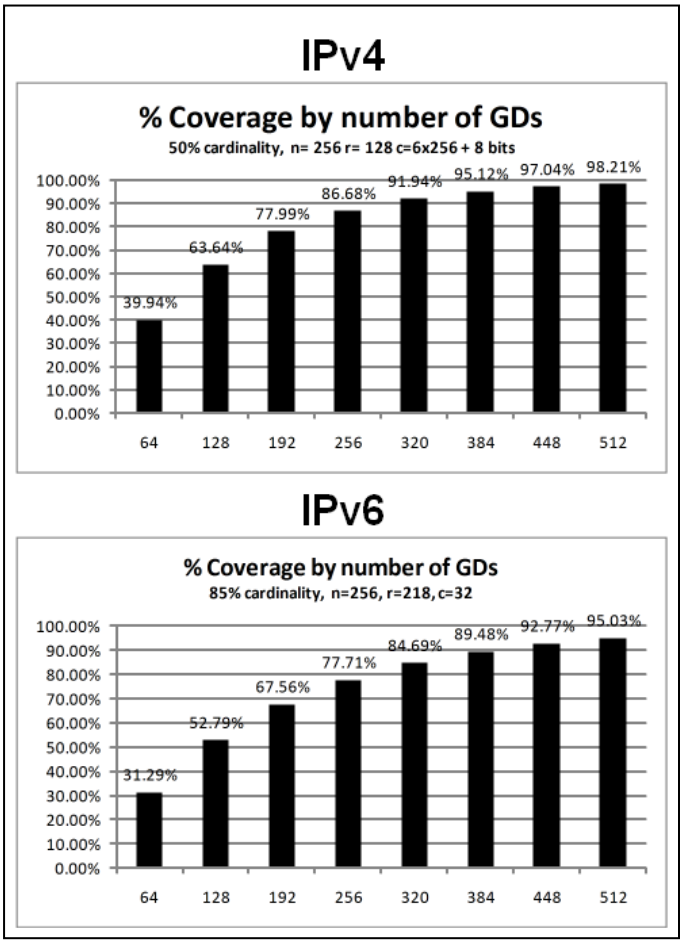

Figure 6. Top graph show high coverage at 512 GDs for an IPv4-connection signature of 7 MFDs at 50% cardinality (half the bits on average turned on in a GD). Bottom graph shows that higher cardinality is required to obtain high coverage with longer patterns such as IPv6 would require.

B. *IPv6 Gang Detector Coverage*

What appears clear is that a GD composed of 7 feature cells, appropriate for IPv4-connection pattern space, has excellent application qualifications. What also appears clear, as shown in Fig. 6 at the bottom, is that there is a limit to how many features can be "practically" accommodated in GD patterns without employing GDs with higher than 50% cardinality.

Other construction parameters for GDs have been probed and tested for effect, and one shows promise for maintaining high coverage (90%+). With a 32 MFD GD, 1024 GDs can cover 90%+ of pattern space at all times. This is likely much more than necessary given the constant refresh of the GDs in service and network inter-endpoint collaboration.

C. *Cardinality and Coverage*

GDs are generated at birth with a biased random generation of feature indicators, whose count is the cardinality at birth. A 60% bias at birth would place 0.6 x 256 = 154 feature indicators randomly in each MFD. The cardinality at birth is thus 154. Training then reduces that cardinality as the GD is tolerized to normal behavior and feature indicators are removed. Diversity among the GDs employed for pattern space coverage is affected by cardinality. If cardinality is too high there is a lot of useless pattern overlap among the GDs that requires processing attention as high multiple hits occur for any given anomaly pattern. If cardinality is too low many more GDs are required in order to cover pattern space completely. It is felt, though not verified, that the optimal working cardinality for GDs in service would be 50%, and that 40% would be an acceptable lower bound.

D. *Testing for GD Training and Service Characterization*

Demonstrating feasibility of gang detector technology with a custom developed simulation system focused generally on IPv4 network traffic, and specifically on in-and-out TCP/UDP/ICMP/Other traffic connections, characterized in packet header data. Scaling to IPv6 connection traffic was analyzed only to confirm feasibility.

Test data sets were generated from weeks to a few months of packet traffic capture from a variety of small-office machines. With a sufficient understanding of how the number of GDs, their structure, and their tolerization training, affects pattern space coverage.

V. Detector Context and Detector Sets

A. *Detector Context in Network Architecture*

An agent, in the sense that we employ the term, is a self contained entity that is responsible for a specific type of detection activity at an endpoint. It is currently anticipated, as shown in Fig. 7, that three agents, of identical gang-detector-based architecture, will each be responsible for a different hierarchical level of detection at each endpoint.

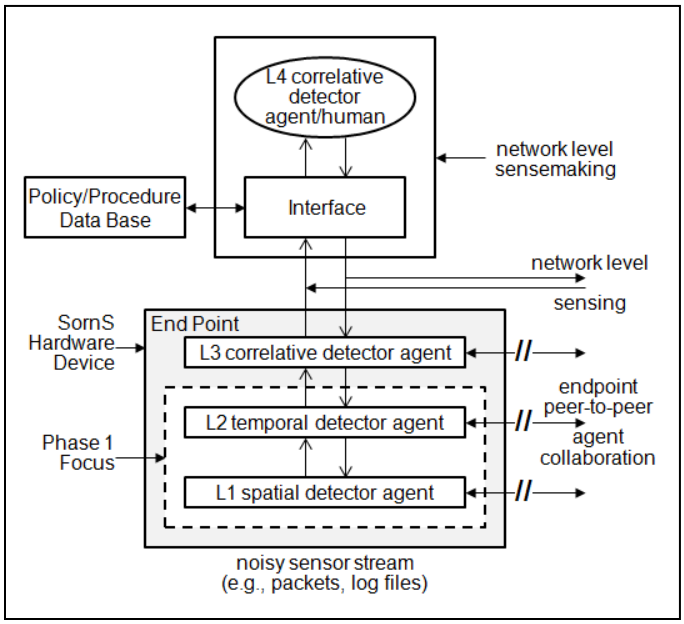

Figure 7. SornS Multi-Agent Endpoint and Network Architecture

Three general types of pattern detection are thought appropriate for the first three levels of SornS sensemaking classification: spatial, temporal, and correlative respectively.

- Spatial detectors at level 1 (L1) are structured to detect contiguous ordered sequences of features: something immediately followed by something immediately followed by something, and so on. The input for L1 in this feasibility study focused on sequential bytes of packets, principally in headers for analytical purposes but also on content for concept extension. L1 sends a feature stream to L2.
- Temporal detectors at level 2 (L2) represent something followed by something followed by something, and so on, but not necessarily contiguous as each contiguous group of features fed to L2 may have been separated from adjacent contiguous features by L1 data considered of no importance. Hence, L2 processes features that are ordered in time and space, all from the same packet but not necessarily spatially congruent. L2 sends a feature stream to L3.
- Correlative detectors at level 3 (L3) are then fed features that represent a temporal sequence of anomalies detected at L2, which may be cross packet and/or cross flow.

The feature-feeding mechanisms between levels and the temporal and correlative detectors, cannot be discussed further in the space limitations of this article.

B. Detector Sets

Fig. 8 depicts the four different detector sets contained within the PatProc for each endpoint hierarchical level, and indicates the life-cycle flow of detectors between sets.

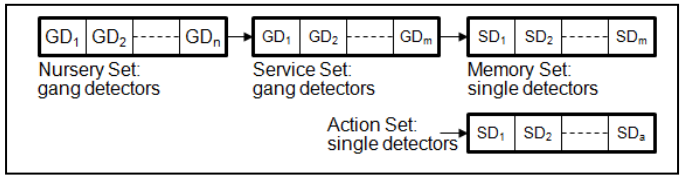

Figure 8. Four sets of detectors filter the same datastream

There are four detector sets, all simultaneously filtering the feature stream presented by the agent's preprocessor:

- Nursery set – this set is where the agent's preprocessor (PreProc) places newly birthed GDs and manages their tolerization process. When a hit occurs during tolerization the agent's postprocessor (PostProc) is notified and removes an appropriate feature indicator from the GD (perhaps multiples) experiencing the hit. When a GD completes its tolerization it is eligible for transfer to the service set. Transfer occurs when there is room for a GD in the service set. GDs in the nursery set are terminated and replaced if they are tolerized below a minimum cardinality threshold. New GDs are added to the nursery set by the agent's GD birthing process whenever there is room.
- Service set – this set contains tolerized GDs that look for anomalies. When one is found the PostProc is notified and accesses the working memory to extract the specific single signature that caused the hit. The extracted signature is then eligible for transfer into the memory set as a single signature detector (SD). Transfer to the memory set occurs when there is room for another SD. Individual GDs are terminated in the service set when their lifetime is over, determined by a combination of total time in the service set and a threshold for minimal time-based detection performance. Time may be measured in either operating time and/or quantity of datastream processed.
- Memory set – this set contains single signature detectors that represent working anomaly detectors. SDs have a utility measure (U) associated with them in working memory. Lifetime for an SD in the memory set is determined by its U value. U values are promoted and demoted by higher level agents according to the utility of the SD in higher level anomaly detection. SDs that fall below a minimal threshold utility value are terminated. A portion of the memory set is reserved for SDs that can be inserted as trials by the agent even when the rest of the memory set is filled with high utility SDs – to help mitigate the effect of environmental changes that might be too-slowly adjusted by higher level utility values.
- Action set – this set contains SDs that if hit will call for or cause immediate action – such as a black list entry for a known bad packet address that then causes the packet to be flushed, or a white list entry that overrides anomaly detection. It is expected that the action set will be populated and depopulated by directives from a level above the endpoint levels. It is not expected that the action set will contain identical network-wide SDs, which are more efficiently dealt with by a network appliance.

Detector sets are associated with specific detection domains. For processing packet data these domains are demarcated separately for packet connection processing and for packet content processing individually associated with a specific application, such as a web application, a Microsoft office application, an SQL server, and such, as shown in Fig. 9. Switching between domains means saving and restoring the state of the relevant detector sets – a no-time cost activity that is part of the on-board PatProc capability, which is also used for immediate switching between multi-packet flows.

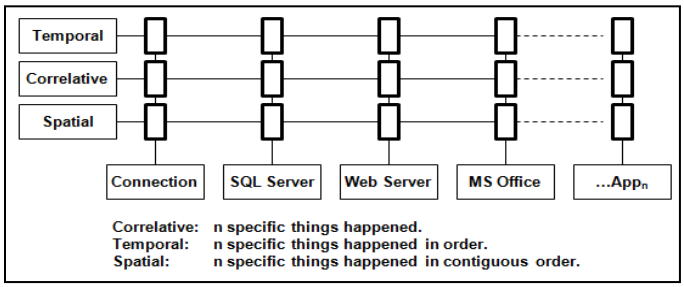

Figure 9. Separate detector sets for each domain of detection

VI. In Conclusion

This project did not begin with any hint that its accomplishments would include a patent-pending Very Large Scale Anomaly Detector (gang detector). Although we had total confidence that the underlying hardware PatProc technology would enable innovative and important advances in high fidelity artificial immune system performance, the breakthrough in anomalous pattern capacity was unforeseen, as was the uncompromising total coverage of a vast pattern space.

Although the gang detector architecture can be usefully implemented in conventional technologies for certain small applications, without the new VLSI PatProc none as scalable and none as inexpensive are known to the author. Some context to keep in mind: the gang detector is good for negative selection, not positive selection, you cannot build a gang detector by adding patterns to it, and you cannot delete a single pattern from it (like Bloom filters that way).

This project focused on the application of the gang detector for network connection-based anomaly sensing, a specific domain instance of the more general "normal vs. anomalous behavior" classification problem. The approach should be self-adaptive to local dynamics and provide custom anomaly detection with no two installations alike in learned pattern content. In summary, analysis for a seven feature packet-connection pattern example showed:

- Memory breakthrough: 193 bytes vs. 10^{16} bytes for pattern storage
- Coverage breakthrough: 512 GDs covers 99.97% of pattern space at 1 endpoint
- Good network-wide coverage does not require high coverage at any endpoint: endpoint multiples boost total coverage with the same coverage curve, and endpoint cyclic refresh boosts total coverage with the same coverage curve over time.

Anomalous behavior detection promises a way round the limitations of looking only for known attack patterns, but it raises new issues in the cyber domain of high false positive rates and questionable stability of normal behavior profiles. Fundamentally, anomaly detectors learn in some fashion what is normal, and then classify all else as anomalous. High false positive rates result from anomaly detections that are benign or of no utility, yet demand human evaluation, as well as normal behaviors that were not present during the training period. The stability issue relates to the dynamics of normal behavior, characterized by changes in personnel and operational behaviors, as well as hardware and software resource changes.

By themselves, the vast coverage of GDs can reduce false negatives, but not false positives, and may increase the occurrence of false positives due to greater coverage of anomalous pattern space. It is anticipated that the ability to reduce false positives will be accomplished by two aspects of the overall architecture: (1) with human evaluation at level 4, principally of correlative anomalies, those that occur in anomalous combinations, will be presented for evaluation, reducing the more numerous quantity of anomalies detected at lower levels; and (2) human evaluative feed-back will be captured as learning in lower-level memory sets and action sets, intending to eliminate repetitive evaluations of the same anomalies.

The issue of normal-behavior stability in typical, but not all, cyber networks will be addressed by at least two aspects of the overall architecture: (1) continuous re-generation of new GDs that will track normal behavior through changes in the environment; and (2) feed-back directives from level four to temporarily suspend certain memory and action set anomaly detection when temporary changes to the environment are made.

The next phase of the project is extending the work reported here further into the hierarchical learning mechanisms, and building endpoint bump-on-the-wire prototypes for operational testing.

Acknowledgment

James H. Burkhard, of Paradigm Shift International, built the simulator, created test sets, and ran repeated tests to probe the operational characteristics of the Very Large Scale Anomaly Detector (gang detector). Lorie DeLorenzo monitored the project and contributed attack test sets for preliminary investigations of packet-content temporal detectors, as part of her Master's project at Stevens Institute of Technology in the School of Systems and Enterprises . This work was funded by the Department of Homeland Security under contract D10PC20039. The content of the material contained herein does not necessarily reflect the position or policy of the Government, and no official endorsement is implied.

References

[1] Dove, Rick. 2009. Pattern recognition without tradeoffs: scalable accuracy with no impact on speed. Proceedings Cybersecurity Applications and Technology Conference for Homeland Security, IEEE Computer Society, March 3-4, Washington D.C. www.parshift.com/Files/PsiDocs/Pap090303-PatternRecognitionWithoutTradeoffs.pdf

[2] Dove, Rick. 2010. Pattern Qualifications and Examples of Next-Generation Agile System-Security Strategies. 44th Annual IEEE International Carnahan Conference on Security Technology, San Jose, California, 5-8 October, 2010. www.parshift.com/Files/PsiDocs/PatternQualificationsForAgileSecurity.pdf

[3] Dove, Rick. 2011. Patterns of Self-Organizing Agile Security for Resilient Network Situational Awareness and Sensemaking. Proceedings 8th International Conference on Information Technology: New Generations (ITNG), April 11-13, Las Vegas, NV. www.parshift.com/s/110411PatternsForSORNS.pdf

[4] Esponda, Fernando, Stephanie Forrest, Paul Helman. 2003. A Formal Framework for Positive and Negative Detection Schemes. IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics.

[5] Forrest, S., S. Hofmeyr, A. Somayaji. 1997. "Computer Immunology." Communications of the ACM 40(10): 88-96. http://cs.unm.edu/~forrest/publications/cacm96-final.pdf

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

[7] Hofmeyr, Steven Andrew. 1999. An Immunological Model of Distributed Detection and Its Application to Computer Security. Ph.D. dissertation, University of New Mexico. www.cs.unm.edu/~steveah/steve_diss.pdf