Symbiot, Inc.
P.O. Box 9646
Austin, TX 78766-9646

A Trajectory for the Evolution of SIMS Architecture

By: Paco X. Nathan, Chief Scientist

Contributors: Mike W. Erwin, Jamie L. Pugh, William W. Hurley

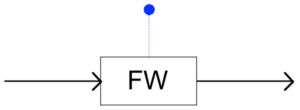
So far in the evolution of security infrastructures, there have been point solutions to point problems: firewalls for filtering packets, virtual private networks (VPNs) for extending intranets, vulnerability checkers for testing hosts, etc. Methods for integrating those point solutions into a consistent framework have been emerging commercially in the form of Security Infrastructure Management Systems *(SIMS). Experience gained from our work in the field underscores how SIMS-based products have been born piecemeal out of necessity, not out of design. They tend to suffer from limiting factors such as unacceptable latencies, being tied to specific vendors and platforms, or other artifactual concerns due to literal descent from earlier computing paradigms, such as* Enterprise Network Management *(ENM). We consider the evolution of SIMS, and present a functional description for a SIMS appliance as the next advance along that trajectory.*

The foundations of network security reach far into the relatively short history of the Internet. Theoretical descriptions for *virtual private networks* (VPN) trace back to papers from the late 1970s, intrusion detection systems (IDS) were first described in 1980, and firewalls have been around[1] since the mid 1980s. Even so, formalisms for coordinated response to network security incidents – and related, commercially available products – did not exist much prior to the Morris Internet Worm[2] on 02 Nov 1988. In the wake of so many hosts and networks being affected, commercially available *point solutions* began to emerge.

The Emergence of Security Infrastructure

Let us consider first about firewalls. Given that the transport mechanisms for TCP/IP networking are built from packets, the simplest way to stop network-based attacks would be to block an attacker's packets. So the original definition[3] for a firewall in 1985 stated: "It is a single point between two or more networks where all traffic must pass". Commercially available firewalls started to become widespread about three years after the Morris worm, when Check Point Technologies introduced *Firewall-1* in 1991.

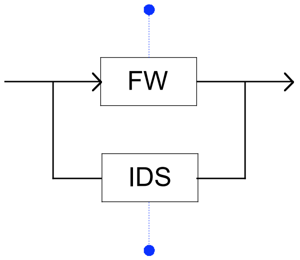
Figure 1 – Diagram for a firewall (1991, Check Point)



The concept of a firewall specifically addresses the threat of malicious packets at a single point, inspecting packets as they cross a network boundary, then *filtering* packets and denying connections based on configurable sets of rules. In the diagram shown in Figure 1, a firewall filters a stream of packets, while also emitting security events to a log file – possibly consumed by a sensor. Assumed but not depicted would be an additional input stream of policy and configuration. Limiting factors for that approach include configurations which allow too much unwanted traffic, or deny too little, constraints on the size of rule sets, and the data quality implications of overwhelmingly large log files.

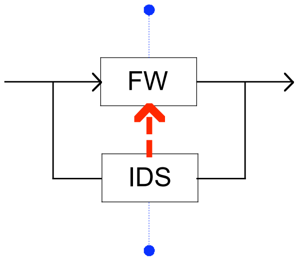
To paraphrase Steven M. Bellovin, one of the original authors to describe a firewall, their point solution creates a barrier between "us" and arbitrarily definitions of "them". While that approach may provide proactive means for security, it begs the question of how does one determine a value for "them" effectively? A study by the US Air Force in 1980 suggested an approach[4] based on *surveillance* techniques. Subsequent implementations[5] produced the first real-time IDS in the mid-1980s, but the devices spread commercially after Internet Security Solutions introduced *RealSecure* in 1996.

Figure 2 – Diagram for an intrusion detection system (1996, ISS)



Similar to firewalls, IDS inspect packet streams, but they apply large sets of pattern rules to monitor for known attack signatures – in effect, attempting to identify "them". Unlike firewalls, IDS send alerts after the fact instead of blocking attacks outright. The limiting factors include excessive rates of false positive and false negative alerts, as well as log files which become far too large for human operators to read. Note from Figure 2 that there are two streams for security events being generated, but no direct communication between the devices.

Figure 3 – Diagram for an intrusion prevention system (2001, NetScreen)

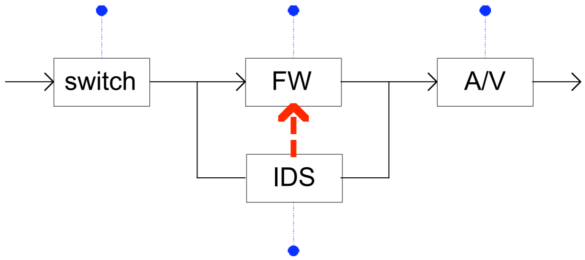


Near the 2001 timeframe, the *structural coupling*[6] between a firewall and an IDS emerged through intrusion prevention system (IPS) products such as *NetScreen*. That innovation completed a cybernetic feedback loop, providing automated means for a pattern matching "expert system" to block potential attacks by regulating a proactive security device. Limiting factors for IPS tend to result from their typical form-factor as a closed box: they will generally be expensive point solutions which do not scale effectively and fail to accommodate input from other sensors – i.e., they tend to be vendor-specific.

Management and Complexity

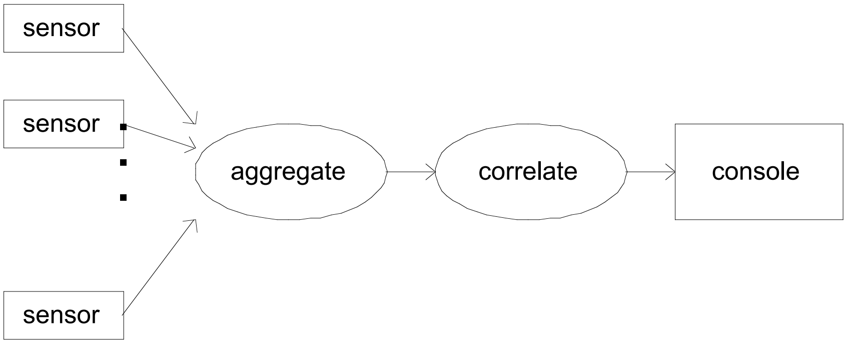
The evolution from firewall to IPS illustrates an increase in the *complexity* of counter-intrusion systems – a trend which seems likely to continue, to keep pace with the evolution in the complexity of attacks. However, packet filtering and signature matching have not been the only point solutions. By the early 2000s many other approaches for intrusion detection and response had become commercially available: network behavior anomaly detectors[7] (NBAD), sophisticated anti-virus checkers, vulnerability scanners, file anomaly checkers, availability monitors, etc. The paired coupling of a simple IPS does not accommodate so many different security approaches – because of the increased complexity of the overall security infrastructure.

Figure 4 – The problem of many sensors poses increased complexity



In 2002, the commercial availability of enterprise security management (ESM) solutions emerged through efforts to *aggregate* and *correlate* security events from a wide range of *sensors*. That is to say, security events coming from a point solution, such as a firewall, get collected and transmitted by a sensor agent to an ESM server, to be aggregated with events from other sensors into a consistent, normalized stream of events. Then the server correlates related events into a logically consistent model of the managed network. For example, consider a connection coming through a switch. The same connection might pass through a firewall, but then generate an IDS alert or an anti-virus warning. The process of correlation establishes relationships within the data, building a real-time model for the security properties of an information infrastructure. In this example, correlation builds a relationship between events related to that particular connection… coming from the switch, the firewall, the IDS, the anti-virus, etc.

Figure 5 – Data flow of enterprise security management (2002, Intellitactics)



ESM solutions accommodate a wider range of sensors than IPS, but they are not necessarily more complex – in fact, they may even be considered *less complex* than some IPS. This illustrates an interesting phenomenon: the hostile environment of the Internet experiences increases in the complexity of attacks – at a particular rate of growth – while the popular measures for countering attacks tend to cycle between increases and decreases in complexity – at their own rates, based on the adoption and maturation of new technologies.

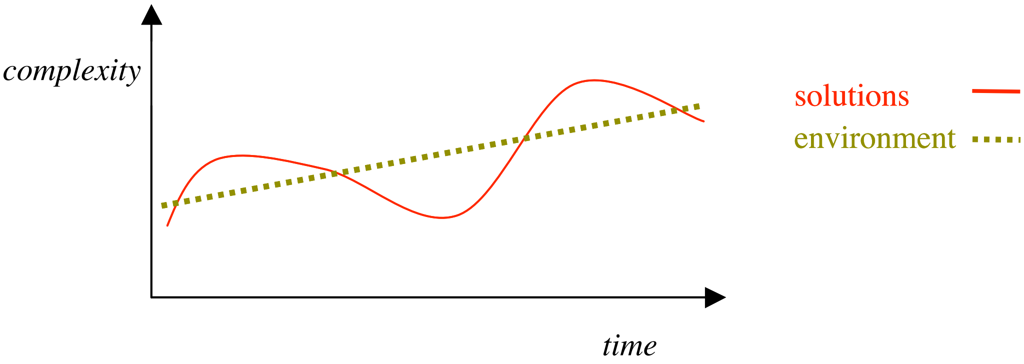


Figure 6 – Growth of complexity for solutions vs. attacks

We see a resulting measure of complexity over time which appears much like a wave moving up a slope. In practice, this phenomenon means that evolving information security solutions will realize relative “peaks” and “valleys” of effectiveness over time. Keep in mind that the organization of a *system*, by definition, must remain more complex than the system’s *environment* if the system is to maintain structural stability. The degree of structural coupling among elements within a system provides a reasonable metric for the system’s complexity. **In this case, the complexity of a security management system must exceed the complexity of its networking environment – or the effective security fails.**

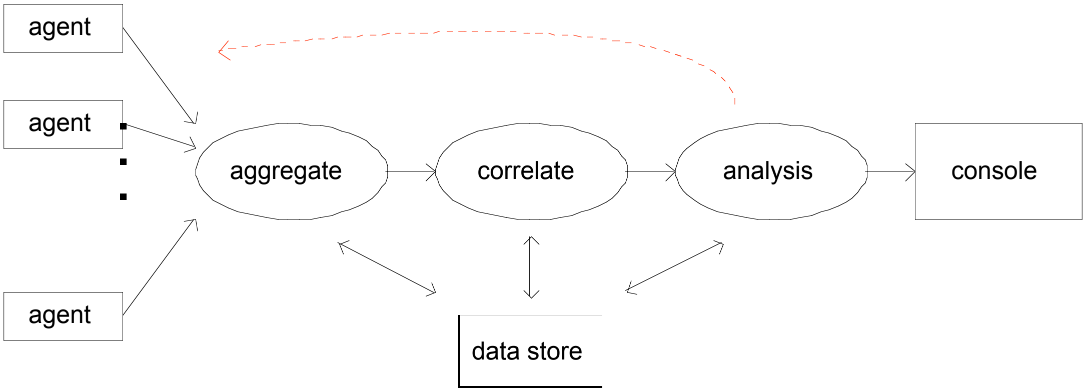
Therein lays the essence of the limiting factors for security management systems: the degree of structural coupling which they engender among security point solutions found in the defended infrastructure limits the sophistication of attacks against which they may effectively defend. For the most part, the early ESM products represented the addition of security event monitoring to well-established solutions for enterprise network management (ENM), such as IBM/Tivoli or BMC/IT Masters – which had become commercially popular since the mid-1990s. The approach pioneered by Tivoli provided *sensors* to collect log files from network components – switches, web servers, database, etc. – sent to the ENM server to be aggregated, correlated, and displayed on a console. The ESM data flow circa 2002 mimics ENM, following an architectural model that is nearly a decade old. To the extent that an ESM product does not engender structural coupling between security point solutions, it provides an expensive console more so than a security management solution.

Table 1 – Security infrastructure evolutionary timeline

1991	FW, VPN, A/V, etc.	security point solutions
1996	IDS	surveillance
2001	IPS	structural coupling for response/regulation
2002	ESM	aggregation, correlation (extends Tivoli model)
2003	SIMS	security infrastructure management

One solution which has evolved in answer to this problem is a *security infrastructure management system* (SIMS). An excellent summary paper is given by Steven J. Scott[8], which describes an idealized SIMS architecture, includes a list of early vendors, and suggests several categories of detailed product evaluation criteria. That approach has gained commercial acceptance circa 2003 through vendors such as Arcsight and netForensics, though independent reviewers[9] appear somewhat reluctant in their praise.

Figure 7 – Security infrastructure management systems (2003, Arcsight)



Important distinctions between ESM/ENM and SIMS result from the fact that SIMS must address the operational difficulties of high network traffic and service demand conditions, like ENM, but must also address the security issues of an evolving hostile environment. Additional requirements may be placed on a SIMS architecture for realizing high-trust and high-assurance operations. The legal, fiscal, and political implications of those requirements – e.g., accessing ambiguity in attack attribution, automating remediation, facilitating forensic analysis, handling evidence, and providing for extensive auditing requirements – create a significantly greater order of complexity for the SIMS problem space. Consequently the material differences between the ESM approach and SIMS can be stated as the latter having "evolved" the additions of:

- A more substantial data store
- An *analysis* process after the correlation
- A more sophisticated console
- Feedback loops to regulate multiple security point solutions

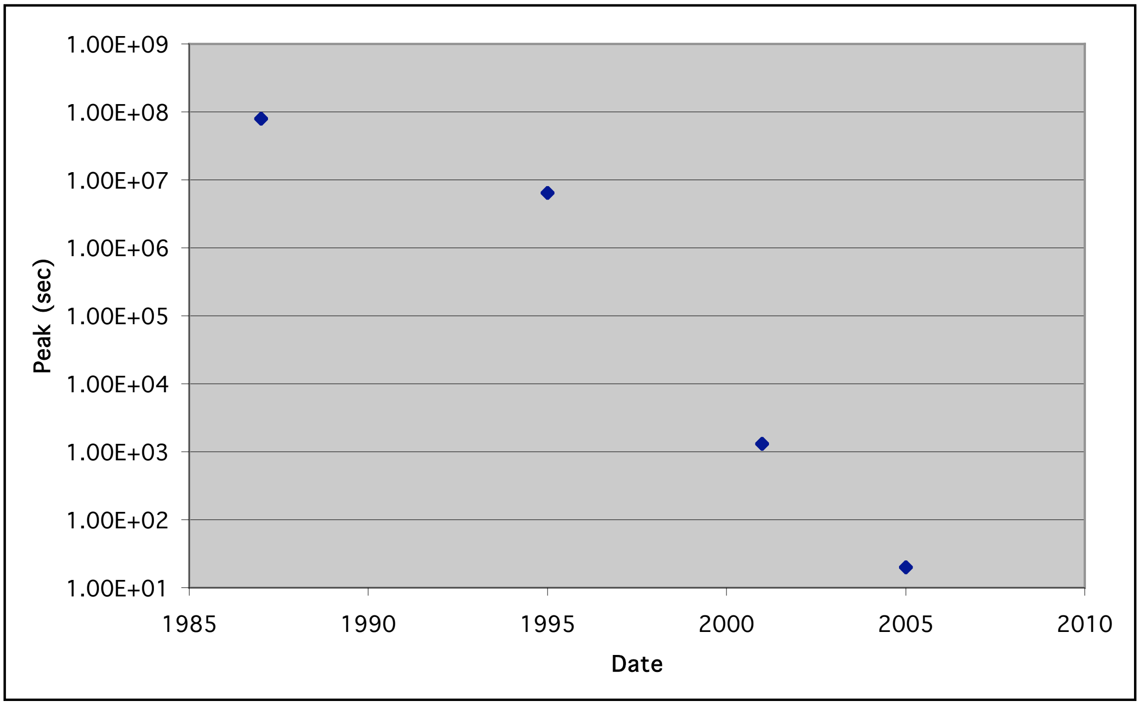
Analysis in this case means that the network model, which results from correlation, gets analyzed to perceive threat and vulnerability – which can then be logged in a database, reported on the console, sent as a notification to some system administrator… or used to regulate security point solutions through *agents*. Agents function[10] either as sensors which measure security properties or effectors which conduct operations. As such, the extent of structural coupling engendered by a SIMS architecture can grow considerably. That allows for an evolution of SIMS, in terms of increased complexity, and hence the potential for effective security even in the face of an increasingly hostile environment.

Limiting Factors

What are possible limiting factors for SIMS solutions? The most prevalent issue is that event latencies become too high. The time interval between the point where a security event gets generated and the point where it influences a critical decision is called *latency*. If the system latency exceeds the time it takes for an attack to inflict damage, then the structural coupling gained by SIMS-based analysis loses significance. In a private 2003 survey of SIMS products, some tended to experience latencies of more than 10 seconds. Recent analysis of worm attacks has shown that the time for peak penetration is decreasing[11] exponentially, and trend-line projections predict that some future worm in the 2004-2006 timeframe will peak throughout the Internet within 30 seconds[12] of initial launch. An enterprise network with a 10 second delay built into its security response would not stand a chance of surviving.

A potential resolution for latency would be to have a SIMS appliance monitor the latencies for its various elements and internal processes, and then use *self-regulation* to compensate for load conditions. That is to say that the more crucial time-dependent processes receive priority based on the critical-path latencies observed during run time. Self-regulation features would effectively build an additional measure of quality assurance into the solution.

Figure 8 – Peak penetration times for malware attacks decreasing exponentially



date	peak (sec)	name
1987	7.884E+07	Friday the 13th
1995	6.480E+06	Concept
2001	1.320E+03	Nimbda
2005	3.000E+01	”Flash” (anticipated)

Another observed problem has been the complexity of installing and configuring a SIMS appliance, i.e. the means by which an appliance establishes an interface with its environment. A potential resolution for installation procedures would be to have a SIMS appliance monitor its networking environment periodically, beginning at boot time – to build and refine a model for the network. We call this process *self-discovery*.

Furthermore, SIMS inherits some problems from its ancestors, such as the fact that false positives are still difficult to manage, or how analysis methods are still based on *a posteriori* approaches like IDS, so that "zero-day" exploits (previously unreported) cannot be modeled effectively.

Keep in mind that SIMS has evolved in response to increasing complexity in the networking environment, and still represents a relatively new approach. In our preliminary investigations, we have researched the available technologies based on SIMS architecture, and our competitive analysis indicates that the current state of the art provides an excellent baseline from which to develop leap-ahead technology solutions, based on many areas of opportunity. Key factors from our analysis include:

- Effective communications among components is still at a very early stage of development.
- Reliance on in-band transport for measuring the security properties of components contributes too much latency to the analysis.
- Attempts to reduce system complexity by encapsulating SIMS features within specific components (smart switches, IPS) will not scale adequately and undermine opportunities for better structural coupling of other components.
- Vendors who enjoy market share for specific COTS components seem reluctant to deploy a SIMS product that accommodates components from other vendors, i.e., to support platform independence.
- Real-time visualizations are mostly limited to the relational database reporting methods prevalent in older ENM approaches.
- Policy tools, decision support, analytics, and inference engines tend to be monolithic and hence abstracted out to the periphery of a SIMS architecture, due to considerations of performance restrictions.
- Control system aspects of both SIMS and ENM emphasize the system administrators, not the decision makers, as primary observers.
- Little consideration has been given toward leveraging a distributed internetworking model for security decision support.

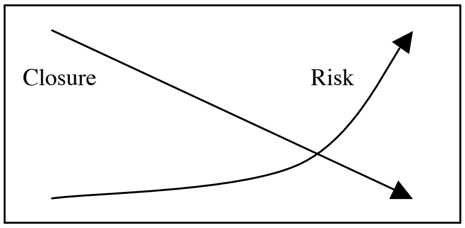
Looking at any given enterprise network, one tends to find many security point solutions in place already: firewalls, managed switches, intrusion detection systems, vulnerability checkers, anti-virus, availability monitors, etc. Prior to the commercialization of a SIMS architecture, the state of the art for correlating and analyzing security events from those components failed to foster much ease of use for defenders who needed to monitor networks under attack and respond effectively. There was far too much security event data being generated during normal operation, let alone during attacks, for an operator to monitor directly. Moreover, the rate of false positives and false negatives inherent in threat detection systems tended to reduce their significance. The system administrators in some cases would ignore security alerts, or face difficulties in being able to follow the context of sophisticated attack vectors, either of which factor ultimately contributes to increased risk and reduced ease of use. That is, effectively, the condition of the IT industry circa 2003.

Cybernetics and Smart Fabric

Considering the security point solutions which one tends to find deployed on an enterprise network, it could be said that much of the *structure* for SIMS has nearly been realized. Most of the necessary components exist. However their *organization* has not been realized. The structural coupling, i.e. the management of communications, among those components is not sufficient for an effective SIMS architecture.

In the context of the theory of autonomous systems[13], we propose a reconceptualization of the SIMS problem space away from existing approaches – focused on literal descent from ENM solutions – and instead **conceptualizing in terms of operational closure with respect to risk**. Consider that an effective, secure information infrastructure maintains some degree of *closure* from threat via well-defined boundaries – e.g. routers, firewalls, etc. – and the *risk* of threat grows exponentially as that closure fails or declines. In that sense, transactions (i.e., services) with other networks represent perturbations of the defended network's boundary. Moreover, effective security can be represented as a dissipation of risk from the system – leading to a potential for modeling security infrastructures as *dissipative structures*.[14]

Figure 9 – Operational closure vs. dissipation of risk



Stated in more general terms, we may consider enterprise network security in the context of second-order cybernetics[15] – i.e., as a "smart fabric" exposing services on the public networks, with a SIMS solution responsible for coordinating structural coupling, running analysis, informing policy decisions, etc. – managing feedback. That "smart fabric" might then be considered as a self-regulating, self-producing system where its topological boundary consists externally as a set of exposed, filtered network services on named hosts within an enterprise network, and internally as another set of exposed, filtered network services on named hosts, plus internal switching/routing capabilities. That morphology must be capable of being perturbed, deformed, and yet maintaining structural stability – which is to say, dissipating risk.

That theoretical basis allows for an existing mix of security devices (heterogeneous, ad-hoc, contingent, isolated) to operate together at a greater level of functional complexity, theoretically capable of being considered ensemble a "cognitive system". Such a reconceptualization would represent a substantial departure from the earlier information processing paradigm of ENM.

Table 2 – Systems-theoretic description of SIMS

space	Public packet-switched data networks.
domain	Peered, autonomous, enterprise networks running on smart fabrics and sharing trust metrics.
topological boundary	A set of exposed, filtered network services on named hosts within an enterprise network, plus potential switching/routing capabilities.
observers	A society which includes the decision makers, e.g. a director or CTO, security personnel, analysts, system administrators, auditors, legal advisors, and related staff.
self-production	Providing availability for the exposed services on named hosts: QoS, NAT, VPN, etc.
self-maintenance	Remediation through dynamic quarantine, rate limiting, shunning, availability monitors, reduction of false positives, etc.
self-reference	Self-discovery, analysis, notification, remediation, data mining, auditing, etc.

A systems-theoretic reformulation helps establish a set of guiding principles for our effort[16], which include the following:

- Diverse security components in a network need to be organized with better structural coupling to realize effective communications and feedback.
- Decision support tools need to be cascaded in a tiered model, corresponding to differing response periods for feedback (see Table 3).
- Decision makers need to be incorporated from a perspective *within* the system, as its primary observers, leveraging a top-down, information design[17] approach for representing visualizations of the real-time changes in defensive posture.

The process of integrating diverse technologies for decision support in a tiered model for producing security response strategy is analogous to implementing a multi-level data cache. Each level depends on policy and decisions from the next, differentiated by response periods, levels of aggregation, levels of abstraction, etc. Perhaps an even better analogy would be the TCP/IP networking stack, where policies and decisions are also abstracted corresponding to different layers.

Table 3 - Tiered model for decision support

tier	**System I/O Fabric**	**SIMS Architecture**	**AI Inference**
expertise	networks	security	intelligence
decision level	low	moderate	high
response period	10^{-2} sec	10^{0} sec	10^{2} sec
features	management protocol adaptive QoS hardware platform out-of-band transport policy enforcement	visualization aggregation risk metrics auditing forensics	analysis remediation plans policy maintenance attack attribution
technologies	linear programming neural networks hardware acceleration	adaptive fuzzy cost estimators rule processing data mining	ontology knowledge base inference planning

Each different tier maintains its own model of the network, using a different level of aggregation and response rates for decisions – being informed by the next higher layer. Thus we use the term "fabric", which otherwise refers[18] to system I/O controllers. In the next generation of SIMS architecture, we foresee the decisions made for security response being coordinated from the level of kernel mods, drivers, I/O bus control, and network interface, upwards… eventually in place of "agents". A combination of realtime metrics and AI inference and planning will drive the reconfiguration of I/O streams within an enterprise network toward remediation through self-production of its topological boundary. The resulting system will provide coupling among the infrastructure components as well as incorporating the distributed consoles used by decision makers.

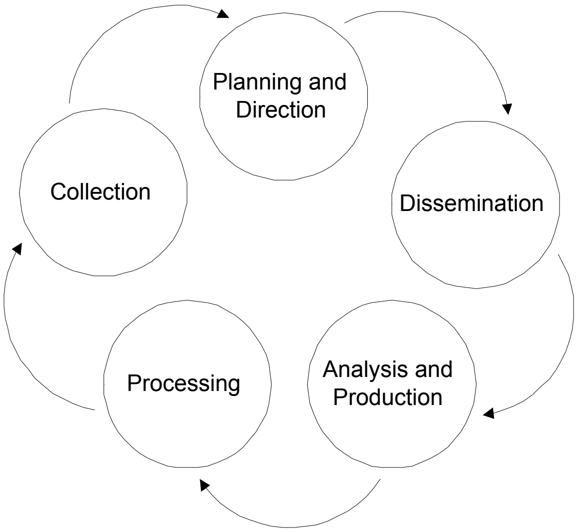
In that sense we take a departure from the older, "information processing" paradigm of aggregating and correlating security events, and rather look toward control systems solutions for self-production, self-maintenance, and self-reference of a smart fabric, based on a tiered, internal model of itself, and defined in terms of a set of external/internal definitions for filtered services on named hosts, plus limited internal switching, which is capable of sustaining deformations. A matrix representing the operational closure of that smart fabric, with respect to risk, determines our basis for assessing, metering, and sharing risk metrics. The sharing of risk metrics could facilitate a form of internetworked structural coupling – more "social" than as a distinct system. The self-maintenance of that morphology is synonymous with the assertion that the information infrastructure security, i.e. SIMS, maintains a greater measure of complexity than its evolving, hostile internetworking environment.

The Intelligence Cycle

Consider how an overwhelming stream of security events must get consolidated, correlated, and analyzed, such that relevant alerts increase in significance and receive appropriate attention. This would seem to be a familiar problem in the domain of intelligence analysis. Whereas the protocols and methodologies of the Intelligence Community have evolved to a substantial degree of complexity and robustness, the underlying protocols for information infrastructure security components are still relatively immature.

Therefore we suggest that the *Intelligence Cycle*[19] be evaluated as an informative metaphor for projecting the evolution of SIMS architecture. In fact, one can make a strong argument for a direct correspondence between the two models.

Figure 10 – Intelligence Cycle



Compare the similarity between the processes of a generalized SIMS architecture in Figure 7 and the Intelligence Cycle in Figure 10. The *aggregation* process of the former parallels the *collection* of the latter, which is defined as "gathering of the raw information needed to produce finished intelligence". Likewise, the *correlation* process of the former parallels the *processing* of the latter, which is defined as "converting the vast amount of information collected to a form usable by analysts". The *analysis* process in both is a direct correlation, defined as "the conversion of basic information into finished intelligence", with emphasis on developing relevance. Finally, the *console* process in the former should ideally correspond to the *dissemination* process of the latter, which is defined as "the distribution of the finished intelligence to the consumers, the same policymakers whose needs initiate the intelligence requirements". Note that in the Intelligence Cycle, the dissemination process feeds into the first process of planning.

Further reading on the subject can be found in an unclassified article[21] by Dr. Michael Warner of the CIA History Staff, which seeks a definition for intelligence. In that work, Dr. Warner explores a 1958 essay[22] by "R. A. Random", notably how intelligence requires some measure of secrecy. Warner goes on to discuss the division between an observer who gathers intelligence and an operative who takes action using it – noting that historically the sophisticated intelligence organizations (KGB excluded) have imposed strong separations between those tasks. As such, the Intelligence Cycle articulates an architecture for a system of feedback and separation of those tasks in varying degrees of secrecy.

Maturana underscores related notions in terms of structural coupling: in our SIMS context, that would correspond to the dynamics of structural coupling between separate *sensor* and *effector* agents, indicating the degree of complexity in the system, and some measure of its degree of intelligence. One might argue that the Intelligence Cycle corresponds closely to second-order cybernetics, specifically to the notion of an autonomous system as defined by Maturana, et al.

Warner concludes by stating his own definition: "Intelligence is secret, state activity to understand or influence foreign entities." Secrecy corresponds in our context with operational transparency. Again, this fits with second-order cybernetics views of cognition, i.e., prioritizing the relevance of the "observer" from within a system. In our context, the notion of understanding and influencing foreign entities could readily correspond to attacker attribution, reduction of false positives, and strategies for remediation.

Conversely, it is interesting to contrast the two models of intelligence and operations, between SIMS and the IC. Since we have identified many points of correspondence, as well as a systems-theoretic basis for comparison, where do those models diverge? Answers to that question point out areas for substantial improvement in the comparatively immature, less sophisticated SIMS architecture. Notably, there needs to be:

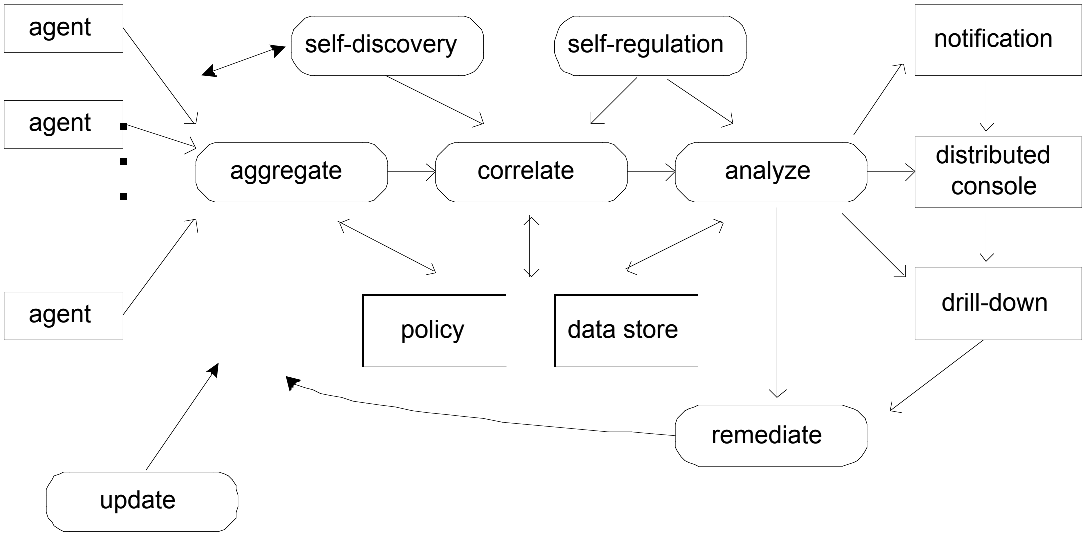
- more process in place for addressing policymaking and planning.
- formalisms for disseminating intelligence, not merely displaying it, to policymakers who are integral *within* the system.
- better means for feedback loops and closure, particularly with respect to policymaking.
- using dynamic aspects of policymaking to guide the agents.

By applying the Intelligence Cycle to problems of high assurance operations for information infrastructures, these are among the issues which we seek to resolve in a next generation SIMS architecture.

Next Generation SIMS

Figure 11 shows a process diagram for an idealized next generation SIMS architecture and the data flow within it.

Figure 11 – Architecture for a next-generation SIMS appliance



To summarize, the essential functions of a SIMS appliance should include capabilities to:

- Aggregate and normalize security event data from a wide variety of devices and monitoring agents into a consolidated stream, taking into account the differences between vendors, host policies, peer networks, etc.
- Correlate events into a consistent, real-time model of the defensive posture of the network.
- Measure the performance envelopes for sub-systems, while identifying limiting factors and enabling self-regulation.
- Analyze the threat, vulnerability, and relative cost of real-time attacks, while planning for remediation options.
- Store the correlated event data and analysis into a database for subsequent reporting, evidentiary handling, forensic analysis, and data mining.
- Send notifications to the appropriate personnel.
- Visualize a real-time representation of the defensive posture on secure, distributed consoles that are suitable for use in a network operations center.
- Provide means for automated and operator-initiated remediation and policy updates.
- Update source code and data for evolving sub-systems through a secure “emerge” mechanism for updating subscribers.

- Assist the decision makers in identifying and reducing the rate of false-positive and false-negative alerts.

To accommodate a wide range of commercial and government environments, the appliance will need to be highly configurable, and amenable to differing levels of classification. Notably, its operational capabilities depend on three streams of information. First, it must depend on an ongoing process of self-discovery for the infrastructure being defended. Secondly, the policy introduced into that appliance will define much of its behavior, particularly its dynamic accommodation of changes in mission and personnel. The definition of policy subsumes many elements: rules of engagement in response to hostile conditions, mission imperatives, personnel status, acceptable use, component-level configurations, etc. Those policy elements will be specific to a particular infrastructure, whether government or commercial, and considered highly confidential and proprietary per application. Thirdly, there are classifications of threat and vulnerability which must be continually updated and imported into the appliance for its dynamic accommodation of evolving hostile environments. The sub-systems likely to be used already interoperate with GOTS standards, such as Information Assurance Vulnerability Alerts (IAVA) and DoD-CERT[23] which are not available to the public. In the case of future commercialization, a next-generation SIMS appliance would have to rely on public or commercial alternatives for such information.

Conclusions

It is interesting to note that the origins of cybernetics emerged from efforts[24] by Norbert Weiner, et al., to combine an anti-aircraft weapon with a rangefinder device for targeting. In a rough sense it was a structural coupling of a counter-intrusion device with an intrusion detection device – i.e. similar to an IPS – which later became decisive technology for defensive weapons systems in the Battle of Britain during World War II.

Our efforts here, to apply the Intelligence Cycle and second-order cybernetics to the problem of coordinating diverse information infrastructure security components, follow a similar design pattern. Attempts to devise a next-generation SIMS architecture as a *smart fabric* can ultimately be viewed as the development of a defensive weapons system for *network centric warfare*[25]. Such systems will gain market share by levering the existing COTS security components which are in place already, and transiting far away from the notion of security management as an "information processing" paradigm.

The next step will be to realize the benefits of this analysis in the form of a next generation SIMS appliance.

Endnotes

[1] Specifically in reference to Cheswick and Bellovin, 1985 – for a more general history, see also Avolio, F.: “Firewalls and Internet Security, the Second Hundred (Internet) Years”, *The Internet Protocol Journal*, Jun 1999, http://www.cisco.com/warp/public/759/ipj_2-2/ipj_2-2_fis1.html

[2] Seely, D.: *A Tour of the Worm*, http://world.std.com/~franl/worm.html

[3] Cheswick, W., S. Bellovin, and A. Rubin: *Firewalls and Internet Security: Repelling the Wily Hacker*, Addison-Wesley, 1994, http://www.wilyhacker.com/

[4] Anderson, J. P.: *Computer Security Threat Monitoring and Surveillance*, 15 Apr 1980, http://csrc.nist.gov/publications/history/ande80.pdf

[5] Bruneau, G.: “The History and Evolution of Intrusion Detection”, *SANS Institute*, 2001, http://www.sans.org/rr/papers/30/344.pdf

[6] Whitaker, R. A.: *Self-Organization, Autopoeisis, and Enterprises*, Association for Computing Machinery, 1995, http://www.acm.org/sigois/auto/Main.html

[7] Henry, T.: “Securing the Enterprise with Network Behavior Anomaly Detection”, Burton Group, 24 Oct 2003, http://www.burtongroup.com/

[8] Scott, S. J.: *Threat Management Systems: The State of Intrusion Detection*, 09 Aug 2002, http://www.snort.org/docs/threatmanagement.pdf

[9] Shipley, G.: “Security Information Management Tools: netForensics Leads a Weary Fleet”, *Network Computing*, 01 Apr 2002, http://www.networkcomputing.com/1307/1307f22.html – and more generally in Hulme, G. V.: “Future Security”, *Information Week*, 25 Nov 2002, http://www.informationweek.com/shared/printableArticle.jhtml?articleID=6504031

[10] Kennedy, C.: *Distributed Reflective Architectures for Adjustable Autonomy*, 2001, http://www.cs.bham.ac.uk/~cmk/pub.html

[11] Anthes, G. H.: “Malware’s Destructive Appetite Grows”, *Computerworld*, 01 Apr 2002, http://www.computerworld.com/securitytopics/security/story/0,10801,69674,00.html

[12] Staniford, S., G. Grim, and R. Jonkman: “Flash Worms: Thirty Seconds to Infect the Internet”, *Silicon Defense*, 16 Aug 2001, http://www.silicondefense.com/flash/

[13] Maturana, H., and F. Varela: *Autopoiesis and Cognition: The Realization of the Living*, Boston Studies in the Philosophy of Science [Cohen, Robert S., and Marx W. Wartofsky (eds.)], Vol. 42, Reidel, 1980, http://web.matriztica.org/555/propertyvalue-6121.html

[14] Prigogine, I.: *The End of Certainty*, 1996, http://www.nobel.se/chemistry/laureates/1977/press.html

[15] Ibid., Maturana and Varela. See also preliminary applications for dynamical systems in network security expressed in Saunders, J.: *A Dynamic Risk Model for Information Tech-*

nology Security in a Critical Infrastructure Environment, 2002, http://www.johnsaunders.com/papers/riskcip/RiskConference.htm and in Dickerson, J., et al.: "Fuzzy Intrusion Detection System", 2002, http://clue.eng.iastate.edu/~julied/research/FIRE/index.html

[16] Winograd, T., and F. Flores: *Understanding Computers and Cognition*, Addison-Wesley, 1986.

[17] See the work of Edward Tufte on "information design", http://www.edwardtufte.com/tufte/ – the notion of this class of interface in discussion is generally called an "executive dashboard", such as from one of those listed at http://www.knowledgestorm.com/search/keyword/Executive%20Dashboard/Executive%20Dashboard

[18] Deierling, K.: "Systems and Infrastructure: Growing pains shape I/O scheme", *EE Times*, 24 Apr 2001, http://www.commsdesign.com/design_corner/OEG20010424S0010

[19] *Intelligence Cycle*, Central Intelligence Agency, http://www.cia.gov/cia/publications/facttell/intelligence_cycle.html

[20] *Intelligence Cycle*, Central Intelligence Agency, http://www.cia.gov/cia/publications/facttell/intelligence_cycle.html

[21] Warner, M.:, *Wanted: A Definition of 'Intelligence'*, Central Intelligence Agency, http://www.cia.gov/csi/studies/vol46no3/article02.html

[22] Random, R. A.: "Intelligence as a Science", *Studies in Intelligence*, Spring 1958, p. 76

[23] See http://www.cert.mil/ and http://www.dodig.osd.mil/audit/reports/fy01/01013sum.htm

[24] Wiener, N.: *Cybernetics: Or Control and Communication in Animal and the Machine*, 1948

[25] http://www.dod.mil/nii/