Online surveillance of computerized systems – Analysis of current and future needs

Mario Couture

Software Analysis and Robustness Group, Defence R&D Canada – Valcartier, 2459 Pie-XI Blvd North, Québec, QC, Canada G3J 1X5 Mario.Couture@drdc-rddc.gc.ca

Abdelwahab Hamou-Lhadj

Dept. of Electrical and Computer Engineering, Concordia University, 1455 de Maisonneuve West, Montréal, QC, Canada H3G 1M8 AbdelW@ece.concordia.ca

Michel Dagenais

Dept. of Computer and Software Engineering, École Polytechnique, P.O. Box 6079, Station Centre-Ville, Montréal, QC, Canada H3C 3A7 Michel.Dagenais@polymtl.ca

Ashvin Goel

The University of Toronto, 2001B Sandford Fleming, 10 King's College Rd, Toronto, ON, Canada M5S 3G4

Ashvin@eecg.toronto.edu

RTO email address

ABSTRACT

The rapid development of software and hardware technologies has led to a significant increase in the number and variety of computer systems and networks supporting command and control (C2) operations. Current operations may use any mix of hosts (any type of computerized system and its software). The risk of occurrence of errors or failures on these hosts has become increasingly larger over the years not only because of the rising complexity of the software and hardware, but also because of the larger number of cyber attacks and their ever-increasing sophistication and diversity. The presence of anomalies in a host may correlate with the presence of important security breaches. Some of these can be very hard to detect and eliminate. They can stay stealthy and dormant for long periods of time, maintaining hosts in a compromised state with the likely consequence of a serious impact on C2 operations when activated.

Current surveillance technologies running on these hosts are relatively limited in their ability to detect unwanted software behaviours and states. Significant improvements in the effectiveness of online host-level monitoring are necessary in order to ensure the dependability of services offered by the hosts during C2 operations. Operators and system administrators need continuously updated reports depicting detected anomalies and their potential impacts in order to be able to build and maintain situational awareness of their hosts, and to be able to react and/or pro-act in timely fashion to correct or prevent any problems. The nature of current and future cyber threats demands detection techniques that are able to cover the widest possible spectrum of anomalies. In this paper, approaches outlined in the "NATO Code of Best Practice for C2 Assessment" (COBP) [1] are utilized to study current and future needs in terms of technologies for online host-level surveillance, which can be considered as another component of C2. We first formulate problems that are specific to this domain and describe their characteristics and constraints according to the COBP prescriptions. An analysis of the various classes of Measures of Merit (MoMs) is then made in order to identify a number of potential solutions for the improvement of host-level surveillance, which could involve both current leading-edge and anticipated detection technologies.

1.0 INTRODUCTION

The rapid evolution of information systems (herein called *hosts*: the hardware platforms and their software), combined with the ever-increasing capacity of communications networks, is the source of new operational capabilities in telecommunications and defence. Levels of technological complexity that have never been reached before by human-made systems are now common, making impossible their full certification. The relatively rapid fielding of software also contributes to yielding systems that contain defects (faults) that were not detected during testing of the final product. Under certain conditions, faults manifest themselves as errors and service (or system) failures [2]. The consequences can be disastrous, particularly if faulty systems support critical operations or infrastructures such as medical, industrial and governmental ones. Effects range from partial loss of performance to complete, sudden loss of services. More subtly, undetected service errors may remain active for a long period of time, producing false results that will be considered (and used as) healthy. A good example is data leakage: a foray by a malicious hacker into an information system may discover and exploit unresolved faults, allowing the installation of hidden advanced malware which could remain concealed and allow the theft of classified data over long periods of time. Anomalies in the system may be the result of design flaws, bad utilization of the information system, or cyber attacks ([3] made a comprehensive survey of anomaly detection).

There is an urgent need to improve significantly the efficiency of the cyber surveillance that is conducted at the host level in order to make possible the online early detection of many more types of unforeseen threats, no matter what their causes are. There is no doubt that the results of this surveillance will represent a solid basis upon which the most appropriate actions (reaction/pro-action) can be identified and activated at both the host and network levels. Currently, the best security and surveillance systems have very limited detection capabilities [4, 5, 6]. Moreover, host- and network-level surveillance systems usually do not interoperate, nor are they able to share the data that would improve significantly the efficiency of detection at both levels. Administrators are in after-the-fact reaction mode, having insufficient information regarding the current degree of dependability of their hosts. Finally, offline forensic investigations suffer from important data gaps that would, if bridged, help to understand past security incidents. This situation is, of course, far from being suitable for critical operations. In this paper, the approaches and principles outlined in the "NATO Code of Best Practice for C2 Assessment" (COBP) [1] are utilized to conduct the *first iteration* of a comprehensive study aiming at identifying the most relevant capabilities for the next generation of *Host-level Force of Surveillance* (H-FoS). Section 2 provides an overview of important causes of problems that may occur at runtime on hosts, and also identifies the corresponding technological gaps. Based on these, classes of Measures of Merit (MoMs) and potential solutions are then identified in sections 3 and 4. Concluding remarks form Section 5.

2.0 THE USE OF COMPUTERIZED SYSTEMS DURING CRITICAL OPERATIONS – SOME CAUSES OF PROBLEMS

2.1 Design faults and normal degradation of systems

Faults within information systems can be grouped into three basic categories: 1- *mechanical faults* (such as hard disk defects), 2- *electronic faults* (in electronic components such as the central processing unit (CPU) and memory chips) and 3- *software faults*. The reliability of hardware components has been the subject of many past studies that have yielded relatively accurate failure prediction models. In this case, reliability is directly

related to length of use (in "active mode" or in "waiting mode"). Electronic components do not deteriorate in the same way as physical mechanisms, even if they pertain to the same type of components. Some of the deterioration mechanisms that were studied are: electro-migration, drift of component physical parameters, transient electrical stresses, excessive heat, and electromagnetic interference [7]. One solution that is often proposed to address the problem of the reliability of physical components is to implement physical redundancy in the architecture.

An important difference between hardware and software components lies in the fact that software does not exist tangibly; it exists only as states and changes of states within electronic components. Thus, software does not undergo the same wear process as mechanical and electronic components. Software errors and failures are mainly caused by unresolved design faults, program bugs, bad configurations, or bad utilization of the system (and cyber attacks, as seen in the next section). Faults in software are (and may remain) present in a component from the beginning to the end of its life. Patches are applied to software components all along their life in order to solve these problems but, as mentioned in the introduction, nowadays the complexity of software makes it impossible to remove all faults. It is important to mention that patches may also add new faults in the patched components.

The probability of occurrence of errors and failures on hosts is a combination of each of the three sources of faults described above, but software faults dominate by far over the other two. Techniques based on source code analysis can be used to evaluate the density of software defects. Studies [8] show that approximately 0.25 fault per thousand lines of source code can be expected when the best engineering practices are utilized (for the best open source projects), and about 45% of these defects can be considered as high-risk. These studies tend to show that the number of defects in software components will continue to increase in the future unless there is a big shift in software engineering practice. This is unlikely because software and hardware components will continue to be more and more complex, and also because the software industry will continue to accelerate the fielding of its software products (reducing further the available test schedule). Other studies show that faults can be found anywhere in the system, including in the core of the operating system (and more particularly in device drivers [9]). Improving the online surveillance of hosts will be increasingly important, particularly if these systems support critical operations and infrastructures.

2.2 Cyber threats and the limits of current security systems

Malicious hackers are well organized and often well sponsored. They have access to very efficient, continuously evolving advanced hacking technologies which are easy to acquire. Malicious activities and malware will continue to be increasingly sophisticated and efficient. In this context, hosts represent interesting entry doors for hackers that want to penetrate large networks. Social engineering and drive-by download are good examples of means that are often used with success [10]. Gathering organizational information, reconnoitring for system faults, and remotely exploiting vulnerable services all lead to the installation of malware of many forms on hosts. Some are: worms, viruses, Trojan horses, spyware, botnets, and rootkits. A major problem is that these can often only be detected at the host level by security systems, which are known to have very limited capabilities (30% detection rate for traditional malware and 24% for Advanced Persistent Threat malware [4, 5]). Moreover, the possibility that a relatively high number of governmental hosts are already compromised by advanced well-hidden malware makes the online detection task much harder. These facts also show that there is an urgent need for deep, refined online surveillance of hosts that is able to face this arms race.

Dynamic host-level security systems can be grouped in two complementary classes: signature-based (also called misuse-based), and anomaly-based. Signature-based detection is limited to known threats for which a signature has been pre-defined. After 20 years of development, signature-based detection is able to do simple

pattern matching and heuristic detection relatively well, but it is still easily evadable through obfuscation and polymorphism because it concentrates on the features of specific malware instances [6, 10]. Moreover, as thousands of new pieces of malware are created every day and since most new signatures demand human intervention, a lag of many days can be expected between the time a malware starts to spread and the time an antivirus can detect it. Signature-based detection engines yield good results in terms of few false positives, but their effectiveness is limited by the number of signatures contained in the database, which is only a small portion of all possible types and variations of undesired behaviours and states in the system.

Anomaly-based detection makes use of a baseline profile defining normal activities (often for file system, log files, network connections, and the kernel). During the detection process, activities that deviate from the baseline are tagged as "anomalies". They may be directly or indirectly related to malicious activities in the system. In contrast to signature-based detection, this detection paradigm tends to generate a lot of false positives because defining a complete and effective baseline profile is not an easy task. On the other hand, it has a better chance of detecting unforeseen attacks such as zero-day and polymorphic attacks.

2.3 Problem summary

Problems related to host-level surveillance can be summarized in the following terms. The nature, complexity and configuration of information systems combined with the highly dynamic and complex environments within which they are used make online surveillance a multidimensional problem. Huge quantities of streaming multivariate data representing software events and states must be analyzed in quasi-real time in order to detect known and unforeseen threats (from any sources) over their large temporal spectra. As these systems, their utilizations and the environments evolve rapidly in time, the data representing inherent software states and behaviours is not stationary, making imperative the continual update of detection techniques and models. Finally, instantaneous degrees of a system's health must be appropriately provided to the operator on duty to allow his building and maintenance of a host-level situational awareness (H-SA; [11] provides an overview of cyber SA). He should have all the information and means to react appropriately in a timely manner to all these threats, ensuring secure C2 operations.

3.0 CLASSES OF MEASURES OF MERIT (MOMS)

Host-level online surveillance activities define a new type of C2 which takes place alongside the military C2: the *Cyber-C2*. In this paper, Cyber-C2 has the same definition as traditional military C2, except that it is applied to the surveillance and protection of hosts at runtime. Cyber-C2 activities observe, analyze and protect military information systems during operations. This section presents the *first iteration* of an analysis that aims to define the next generation of *Host-level Force of Surveillance* (H-FoS), which should fully support Cyber-C2 (addressing the problems described in Section 2). Approaches defined in the NATO Code of Best Practice (COBP) [1] are used to identify and structure a number of classes of Measures of Merit (MoMs), which could be used to characterize the performance of Cyber-C2. To do that, high-level mission elements of the H-FoS and corresponding classes of measures of force effectiveness (MoFEs) are identified. Classes of measures of C2 effectiveness (MoCEs) are then deduced from the identified classes of MoFEs. MoCEs are metrics that characterize the impact of the H-FoS on the Cyber-C2 context. Potential solutions are then proposed in Section 4. As this paper only presents the results of the first iteration of the COBP process, the identified MoMs are presented under the guise of *classes* (a class being a group of related metrics). Also, the focus of this iteration is on Cyber-C2 as applied at the host level, and only the first two phases of the OODA loop process are considered (*Observe*, *Orient*, Decide, Act). Table 1 lists the two high-level mission elements

(1, 2) that are proposed in this work, and the corresponding classes of MoFEs (1.x, 2.x). In this iteration, the first and second mission elements are limited to 6 and 3 classes of MoFEs, respectively, that in turn are limited to a total of 20 and 5 classes of MoCEs, respectively (1.x.y, 2.x.y; Table 1).

Table1. High-level mission elements and classes of Measures of Merit (MoMs).

High-level mission elements of the H-FoS	**Classes of MoFEs** *How the H-FoS performs its mission or the degree to which it meets its objectives. Object of interest:* ***the H-FoS as a whole***
1 Efficiently support system administrators in charge of host-level Cyber-C2	1.1 Detects any type of potential problem on the host at runtime 1.2 Prevents the propagation of detected problems (local, global) 1.3 Supplies adapted reporting for system administrators 1.4 Supplies adapted control of the H-FoS (man., auto., mixed) 1.5 Evolves with respect to new technologies and threats 1.6 Contributes to maintaining the dependability of the host
2 Allow online and offline collaboration between all types and levels of Cyber-C2	2.1 Is a harmonized whole made of collaborating systems 2.2 Collaborates with other types of FoS (interoperability) 2.3 Supports external (offline) analyses of data and software
Classes of MoCEs *The impact of the H-FoS within the Cyber-C2 context. Objects of interest:* ***the systems of the H-FoS***	
1.1.1 Detect all types of anomalies at the host level (known, unknown) 1.1.2 Detect over flexible ranges (from specific components to holistic, large temporal spectra) 1.1.3 Detect anomalies in a timely manner (early detection; as anomalies appear on the host) 1.1.4 Yield results that are complete, precise and with very low false positive rates	
1.2.1 Enable the host system to be self-adaptive and self-reconfigurable to react to very fast acting/propagating errors/failures/threats 1.2.2 Holistic anomaly detection and analysis (involving all software components of the host)	
1.3.1 Report anomalies in a timely and appropriate manner (using graphical user interfaces) 1.3.2 Supply adapted reporting of the H-FoS (supporting different levels of expertise) 1.3.3 Efficiently support the building and maintenance of host-level situational awareness (H-SA) regarding the instantaneous degree of dependability of the host and possible courses of actions 1.3.4 Maintain the level of trust of the officer in his host (and the H-FoS) 1.3.5 Supply adapted training (supporting different levels of expertise, different roles)	
1.4.1 Supply complete adapted control of the H-FoS (supporting different levels of expertise) 1.4.2 Allow systems of the H-FoS to be controlled locally or remotely 1.4.3 Allow systems of the H-FoS to operate in manual, autonomous or mixed modes 1.4.4 Support the online awareness and control of the degradation of the host's dependability	
1.5.1 Are highly scalable and maintainable 1.5.2 Are easily upgraded or updated when the host and its software are modified (i.e. patches) 1.5.3 Learn to manage new unforeseen threats, as well as normal human interactions with H-FoS 1.5.4 Are tamper-proof	
1.6.1 Ensure the high availability, confidentiality and integrity of the host (services and data) through new detection techniques, models and architectural patterns	
2.1.1 Ensure a synchronized and harmonized whole entity (systems work as a harmonious whole) 2.1.2 Self-optimize through self-reconfiguration to maximize/minimize detections/false alerts	
2.2.1 Interoperate/collaborate with Network-level Force of Surveillance (N-FoS)	
2.3.1 Save appropriate data for the continuous offline improvement of information systems 2.3.2 Efficiently save the appropriate data in a timely manner for offline forensic investigations	

The first mission element of the H-FoS is to support the system administrator in charge of host-level Cyber-C2. To do that, the H-FoS should ideally be able to detect all types of problems that may be triggered on the host at runtime (MoFE 1.1). They may originate from design flaws, bad configuration or utilization of the systems, or cyber attacks. The H-FoS should put into action the mechanisms and models that will allow the early detection (MoCE 1.1.3) of known and unknown anomalies in the information system (1.1.1), with acceptable false positive rates (a major problem with most Host-based Intrusion Detection Systems; HIDS) (1.1.4), anywhere on the host (1.1.2).

Administrators must receive accurate information regarding detected problems (1.3) in order to react/pro-act appropriately (1.2 through 1.4), quickly achieving suitable effects on the host (e.g. limiting data leakage or loss of performance or services; 1.6, 1.6.1). They must be informed in a timely manner using the appropriate level of detail (adapted to their degree of expertise; 1.3.1, 1.3.2) in order to build and maintain host-level situational awareness (1.3.3, 1.4.4), and to create a high level of trust of the administrator in his host (1.3.4, 1.6). Considering the sophistication, distribution and timeframe of cyber threats, the H-FoS should be able to work autonomously (ability to address quick-acting threats by itself), manually (with human intervention), or a hybrid of the two (1.4.1, 1.4.3). H-FoS should provide the means for secure local or remote control (1.4.2).

The ability of the H-FoS to evolve with respect to new, unforeseen threats is also very important (1.5). The H-FoS should be able to quickly adjust the focus and resolution of the observations made within the information system (1.1.2) in order to self-adapt (2.1.2) to very fast moving/acting/propagating threats (1.2.1, 1.2.2). Ideally, it should be able to learn to recognize and manage new unforeseen threats through their effects and then figure out how to detect them earlier (1.5.3). The H-FoS should also offer sufficient scalability in order to harmoniously integrate new leading-edge detection technologies as they become available (1.5.1). It should also easily adapt its models to accept any legitimate modifications made to the information system (software patches [12]; 1.5.2). Collaboration (mission element 2) is another important mission of the H-FoS, which must be able to: 1- work as an efficient harmonious whole to maximize the detection process and minimize resource utilization (2.1.1); 2- collaborate with other types of forces of surveillance (such as N-FoS; 2.2.1); and 3- provide the necessary support for offline analyses (such as forensic investigations; 2.3.1, 2.3.2).

4.0 THE NEXT GENERATION OF H-FOS – POTENTIAL SOLUTIONS

Based on the identified classes of MoMs[1], a number of *potential solutions* for the H-FoS are identified and briefly described in this section. Together, these constitute the vision of an *integrative framework* (Figure 1). Collaborative R&D efforts are needed to push further the development of new advanced detection techniques/models; some are already available but many gaps remain to be addressed.

The proposed H-FoS is made of three concurrent *lines of surveillance* between which components are able to exchange data and to make requests (controls) of each other. The first line of surveillance is constituted of security systems and data capture tools that provide the raw data to a number of detection techniques/models in the second and third lines. Techniques/models can be recombined and reconfigured at runtime in many different ways according to the situation, giving the H-FoS the highest possible *detection dimensionality*[2]. This detection dimensionality allows the H-FoS to be a self-adaptive system that is able to continually

[1] In this section, references to identified classes of MoFEs and MoCEs are omitted for clarity.

[2] *Detection dimensionality* refers to the number of available combinations of techniques and models that can be put into action at runtime.

optimize the detection process and minimize false positive rates based on results of detection analyses and lessons learned from past incidents.

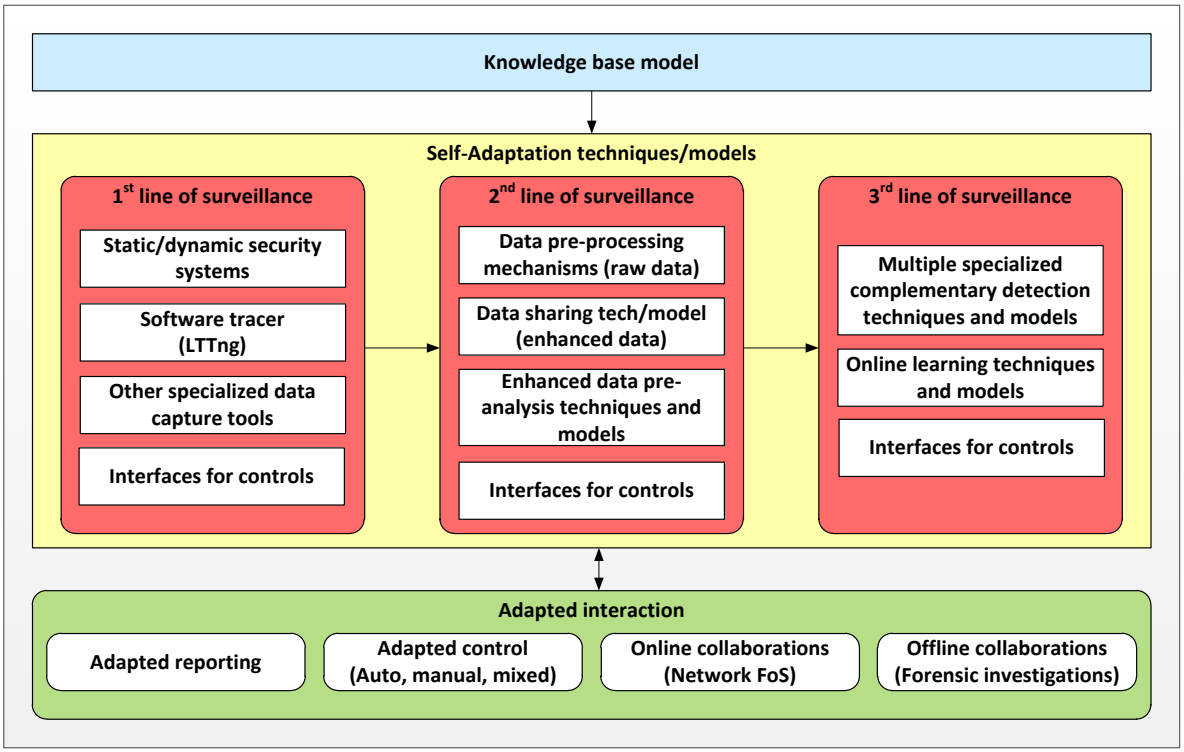

Figure 1. Overview of proposed solutions put forward for the H-FoS.

The knowledge base model defines healthy software states/behaviours for the whole system. This model, combined with the many techniques/models of the second and third lines of surveillance, makes possible the health-based holistic detection of errors or failures that span many components of the system. The modular self-adaptive architecture of the H-FoS (made up of loosely-coupled modules) will minimize the risk that one of its components could stop the H-FoS functioning at runtime (e.g., due to a cyber attack for example). Most detection activities in the second and third lines of surveillance can take place on the host or be distributed (decentralized) over the network. The H-FoS is also able to interact in many ways with humans and external systems (Adapted interactions; Figure 1).

First line of surveillance (sensor data collection). Many efficient technologies are already available on the market for the first line of surveillance. As mentioned in Section 2, current signature- and anomaly-based security systems have important limitations but should nevertheless be integrated with the H-FoS, particularly new leading-edge detection systems as they reach fielding maturity. The first line of surveillance is thus composed of the following mixed configuration of security and observation systems: signature- and anomaly-based security systems (such as antivirus, host-based intrusion detection systems; HIDS), static rule-based security systems (such as firewalls), software tracers and profilers, and other specialized data acquisition modules that are able to peek at specific system state variables. Examples of data types that are generated include: alerts, execution traces, resource usage statistics, and the values of software variables or states in the system. Together they represent a relatively rich set of events and states for analyzing the system. For example, the software tracer LTTng [13, 14] offers an important advantage for the self-optimization of H-FoS.

It allows activating/de-activating (locally or remotely) static and dynamic tracepoints anywhere in the system. This online control of LTTng probes allows the selection of both the focus and resolution of trace events for further detection analysis. Combined with in-system peeking, this online feedback-directed control of LTTng probes represents an important lever which self-adaptation techniques should use to continually improve the quality of detection analyses that are made in the second and third lines of surveillance.

Second line of surveillance (entity and situation assessment). The second line of surveillance captures, fuses, stores and pre-analyzes the *raw data* at runtime. Some studies have started to extend the "traditional" data fusion framework for the cyber domain [15, 16 among others]. The idea here is to adapt and push further this technology in the context of host-level Cyber-C2: the continual online production of *enhanced data* through the fusion of raw data that originates from multiple sensors on the host. Data normalization (cleaning, automatic alignment of semantics, identification of common reference points) should lead to the continual production of enhanced data, which should be captured and saved into a *shared data store*. Some work is currently under way to develop a data model that is able to efficiently capture and semantically link huge amounts of execution trace data at runtime [17]. This new technology should be extended to strive for efficient multivariate data storage and allow for as much pre-analysis as possible of all stored enhanced data.

Continual pre-analysis of these stored enhanced data should aim to: 1- identify new *identities* (e.g., anomalies) on the host and relationships between them (through correlation and cross-correlation analysis for example), 2- estimate the attributes of newly found entities (possible malware involved, local sources of threats, possible intent), and 3- pre-classify entities. This pre-analysis of the enhanced data proactively identifies elements of information that will contribute to further improve the more detailed detection analyses of the third line of surveillance. It should be noted that online discovery of true positive anomalies on the host is a gradual process that involves running many complementary detection mechanisms (both in parallel and in series), which are reconfigured according to the situation. Pre-analysis made in the second line of surveillance is more holistic because enhanced data may relate to any software component on the host. Cross-correlation analysis will thus be able to find malware that spans many software components of the host.

Third line of surveillance (health-based anomaly detection). As mentioned in Section 2, host-level anomaly detection is a complex non-linear multidimensional problem that involves searching for undesired software behaviours and states anywhere in the system, the telltales of which may be spread over timespans stretching from milliseconds to many months. Clearly, the coverage and efficiency of detection technologies and models in the first two lines of surveillance are not sufficient to address this problem. The third line of surveillance does this by increasing significantly the *detection dimensionality* of the H-FoS: the concurrent use of many specialized complementary detection techniques that involve models of the expected healthy behaviours and states of the host at runtime. A higher dimensionality allows higher *variability* (as defined by [18]): the number of distinct actions (related to defined classes of MOFEs) that can be taken at runtime in a specified interval of time. The third line makes use of (and controls) the data and information that are generated in the first two lines and is able to peek anywhere in the system for values of specific software variables or states.

A review of the scientific literature [19, 3] reveals that there are many ongoing efforts to identify new efficient detection techniques/models from domains like Statistics, Machine Learning, Data Mining, and Artificial Intelligence. But, few are sufficiently mature to be fielded yet. Moreover, very few of these studies are dedicated to host-level anomaly detection and, among those, most are done offline and only involve system call events. Holistic detection analysis should not be restricted to system calls to identify traces of errors and

failures that span over many components of the system since some cyber attacks do not require the execution of any system calls [20]. Kernel rootkits, for example, can execute entirely within the kernel context without issuing any system calls once the exploit injects the rootkit. Thus, system-call-based monitoring methods would not detect these attacks. Furthermore, the exploit injection may not require the execution of any system calls in order to be successful. The host should be studied and analyzed according to all its software components (both in the user and kernel contexts) and according to the inherent logical software layers.

It is clear that extensive comparative studies must be conducted to identify the best techniques/models for this line of surveillance. The domains of analysis of each technique must be compared in order to identify overlaps and gaps. Usability of such techniques and models at runtime should be evaluated. The specifics of each monitored system, its utilization in its environment, and the constraints imposed by observational data must be considered as well. For example, the states and behaviours of a system may evolve with respect to time (e.g., upgrades of the monitored host), making observational data relatively non-stationary. Detection techniques based on pre-defined statistical models (which focus on the process that generates observational data) need to be updated on a regular basis [12]. Also, well-trained hackers may be capable of training statistical anomaly detection systems to accept abnormal behaviours as normal.

The techniques and models from Machine Learning focus on building systems that improve their performance based on past results. They appear to be appropriate for online detection of unforeseen anomalies in the system. Models of each technique will be trained offline in the laboratory and then fielded in the H-FoS. Techniques that are currently under study in different laboratories are: *Misuse/signature-based detection* (Rule-Based, Artificial Neural Network (ANN), Support Vector Machine (SVM), Genetic Programming, Decision Tree, Bayesian Network) and *Anomaly-based detection* (Rule-Based, ANN, SVM, Nearest Neighbour-Based, Hidden Markov Model [21, 22], Kalman Filter, Clustering, Random Forest, Principal Component Analysis, Information Theoretic) [19, 23, 3]. Other types of techniques should be studied as well. For example, wavelet transformation and system identification theory [24] could be adapted to the host-level detection context. Ideally, most of these techniques and models should be able to continually learn and improve from normal utilization of the system and security incidents.

Each potential technique/model presents advantages and disadvantages that must be closely scrutinized to identify the *how*, *when*, *why*, *where* and *who*. Moreover, the possible integration and harmonization of many of these [19] should yield a pool of many specialized *hybrid configurations*, increasing the detection dimensionality and thus the detection capability[3]. This pool will contribute to help the H-FoS to self-adapt to different situations: the continual self-optimization of the detection process through the online selection and execution of the best predefined specialized hybrid configuration.

Knowledge base model. Health-based anomaly detection, short-term inferences, risk, impact and damage assessments all imply the utilization of a knowledge base model that defines healthy or normal software behaviours/states that can be expected at runtime. This model should cover all software components and interrelationships between them to allow for holistic detection analyses. The model should facilitate automatic online reasoning by providing the information that will help with the diagnosis of specific known and unknown anomalies detected in the system, with the automatic categorizing and framing for root causes, and with side effects discovery and understanding.

[3] For example, [25] have shown that Bayesian network classification of events that originate from many complementary HIDS models improves the aggregation of the different model outputs, producing less false positives.

The difficulties of modelling the knowledge of complex information systems lie in the huge number of software components and interrelationships between them, which leads to huge quantities of software behaviours and states. Traditional modelling techniques, like finite state machines and rule-based expert systems, would yield knowledge models that would be impracticable at runtime. Nevertheless, some of these traditional techniques can be efficiently used to model a selected number of critical dynamic activities that take place in the kernel. For example, they can efficiently be used to detect hooks (at the levels of descriptor tables, drivers I/O, system entry points) and hidden undesired objects in the system (such as processes, threads, files, key data, alternate data streams, direct accesses to the kernel, etc.) The detailed modelling of the system health (on larger scales) should involve very efficient non-linear techniques that are able to consider many software components. Many techniques from Machine Learning and Deep Learning used in the third line of surveillance should be used to build and train models based on the utilization of source codes and healthy execution traces as input. The Knowledge base model should be made of many integrated complementary models whose domains of analysis overlap.

Self-adaptation (optimization). Self-adaptation of the H-FoS is the self-optimization[4] of the detection process with respect to new situations. Self-optimization is achieved through the selection and execution of the best hybrid configuration that will maximize anomaly detection and minimize false positive rates. Autonomous H-FoS will be self-aware and keep track of the state and quality of the detection process, and will know how to choose among the best hybrid configuration options when changes are observed. This process represents an important challenge that is not completely solved at the moment. Humans will thus have to be involved until new advanced control techniques and models can make the H-FoS autonomous. A limited number of pre-defined hybrid configuration options should be identified first.

To illustrate this process, we give the example of a HIDS that has triggered an alert that is considered as highly uncertain but potentially dangerous for the host. Immediate actions (of the autonomous H-FoS) would be to: 1- activate the appropriate number of focused LTTng probes to produce better enhanced event data, 2- peek directly in the system to get the missing values of software states, 3- identify some appropriate hybrid configuration options, and 4- execute the best configuration to produce better results. To do this, the H-FoS should be aware of the specific components of the system that are directly and indirectly involved in this alert, their normal healthy states and behaviours (from the knowledge base model), and critical locations in the software from which events (from the tracer) and states (from in-system peeks) should be generated.

Self-adaptation techniques and models should be able to keep track of (and learn from) past and current utilized configurations (detection techniques and focus of detection analysis) as well as the evolution of the preciseness of obtained results. Newly detected anomalies will provide guidance concerning what should be analyzed and the knowledge base model will guide the selection of probes to switch on and of detection mechanisms to reconfigure. The ideal type of self-adaptation would be *goal-seeking* or *purposeful* [26]: the use of variable and chosen means that are brought into play in both competitive and collaborative modes for variable and chosen finalities. Techniques from Reinforcement Learning (among others) might have to be considered for the self-adaptation (self-optimization) of the H-FoS.

Adapted interactions. Interactions of the H-FoS with humans, the network FoS and offline deeper analyses such as forensic investigations should be made possible by the H-FoS (Adapted interactions; Figure 1). The H-FoS is well positioned to provide the best data, information and knowledge for all these contexts.

[4] Self-healing and self-protection of the H-FoS are not considered in this first iteration of the COBP process.

Interoperability frameworks such as the now old LISI [27] (among others) could help define and structure how H-FoS will interoperate with network-level FoS. Graphical user interfaces (GUIs) should be adaptable to different levels of expertise of the operator on duty, which may range from novice to highly-qualified. The quality of the transmitted information combined with appropriate and efficient GUIs and machine-to-machine interfaces should lead to the most appropriate cyber reactive and proactive actions at the level of both the host and network. The completeness and timeliness of the data that will be saved on disk for offline forensic investigation should ideally allow for court-quality analysis of security incidents.

5.0 CONCLUDING REMARKS

The approach defined in the NATO COBP [1] was used to conduct *the first iteration* of an analysis aiming at identifying concepts, techniques and models that could define the next generation of Host-level Force of Surveillance (H-FoS). Problems and technological gaps related to online surveillance of hosts were first defined and used to identify many classes of Measures of Merit (MoMs) that are needed to characterize the performance of a new type of C2, the Cyber-C2. The identified classes of MoMs were then used to deduce potential solutions that could be implemented in the next generation of H-FoS, which is well aligned with the NATO Cyber Defence Capability Framework [28]. The proposed H-FoS is constituted of three concurrent *lines of surveillance* within which interoperable techniques/models are used together (according to their detection dimensionality) to better address the multivariate detection problem. The *knowledge base model* allows holistic health-based detection, while *self-adaptation* automatically synchronizes and reconfigures detection modules for self-optimization. *Adapted interaction* modules ensure appropriate exchanges of data and information with the operator on duty (Adapted Reporting and Control) and other FoS systems (such as Network FoS), and save the most relevant data for offline forensic investigations.

The proposed solution is a framework within which a number of current technologies can be used, while others remain to be developed to bridge gaps. The LTTng software tracer, HIDS, firewalls, and possibly some antiviruses can be used as-is in the first line of surveillance. Techniques and models from the Data Fusion domain and specialized data modelling/storing can be adapted to the context of host-level anomaly detection (in the second line of surveillance). The same is true for Machine Learning and other techniques in the third line of surveillance. The knowledge base model, self-adaptation techniques/models, and the integration (harmonization) of detection techniques/models will demand more R&D efforts, which may be eased by evolutionary engineering [18]. The next iterations of the COBP process should put more emphasis on these problems and define an open standard that would facilitate the integration of H-FoS components that originate from different organizations/countries. The possible utilization of redundancy combined with diversity in architecture (at the host level [29, 30]) should also be investigated more in depth to improve the high-availability and resilience of the host, as well as the online detection of very hard to find anomalies caused by unforeseen advanced cyber threats.

REFERENCES

[1] NATO Code of Best Practice for Command and Control Assessment. North Atlantic Treaty Organization (NATO)/U.S. Department of Defense (DoD) Command and Control Research Program, ISBN 1893723097, 2002.

[2] A. Avizienis, J.-C. Laprie, B. Randell, and C. Landwehr. Basic Concepts and Taxonomy of Dependable and Secure Computing. Dependable and Secure Computing, Vol. 1, No. 1, 2004.

[3] V. Chandola, A. Banerjee, and V. Kumar. Anomaly detection: A survey. ACM Computing Surveys (CSUR), Vol. 41, No. 3, 2009.

[4] Mandiant. M-Trends – The Advanced Persistent Threat. Mandiant M-Trends Report, 2010.

[5] Bell. A Study for the Public Security and Technical Program. Bell Confidential Report CPQ0229.1.36.4, Bell Canada, 2010.

[6] J. Viega. The Myth of Security. O'Reilly, ISBN 9780596523022, 2009.

[7] P. O'Connor. Practical Reliability Engineering. Wiley, 4th edition, ISBN 0470844639, 2002.

[8] SCAN. Open Source Reports. http://scan.coverity.com/report (accessed Feb. 2012), 2009 & 2010.

[9] A. S. Tanenbaum, J. N. Herder, and H. Bos. Can we Make Operating Systems Reliable and Secure? IEEE Computer Society, Computer, Vol. 39, No. 5, 2006.

[10] M. Egele, T. Scholte, E. Kirda, and C. Kruegel. A Survey on Automated Dynamic Malware Analysis Techniques and Tools. ACM Computing Surveys (CSUR), Vol. 44, No. 2, 2012.

[11] G. P. Tadda and J. S. Salerno. Overview of Cyber Situational Awareness. Springer, Advances in Information Security, Vol. 46, ISBN 9781441901392, pp. 15–35, 2010.

[12] M. E. Locasto, G. F. Cretu, S. Hershkop, and A. Stavrou. Post-Patch Retraining for Host-Based Anomaly Detection. Department of Computer Science, Columbia University, CUCS-035-07, 2007.

[13] http://lttng.org (accessed Feb. 2012).

[14] M. Dagenais. Efficient Surveillance of Information Sytems Online. Accepted: CrossTalk, 2012.

[15] N. A. Giacobe. Application of the JDL Data Fusion Process Model for Cyber Security. Multisensor, Multisource Information Fusion: Architectures, Algorithms, and Applications, SPIE, Vol. 7710, 2010.

[16] I. Corona, G. Giacinto, C. Mazzariello, F. Roli, and C. Dansone. Information Fusion for Computer Security: State of the Art and Open Issues. Information Fusion, Vol. 10, No. 4, 2009.

[17] A. Montplaisir. Stockage sur disque pour accès rapide d'attributs avec intervalles de temps. Mémoire de maîtrise, Polytechnique de Montréal, 136 pages, 2011.

[18] Y. Bar-Yam. Engineering Complex Systems: Multiscale Analysis and Evolutionary Engineering. Springer, Complex Engineered Systems, New England Complex Systems Institute, pp. 22–39, ISBN 3540328319, 2006.

[19] S. Dua and X. Du. Data Mining and Machine Learning in Cybersecurity. CRC, ISBN 9781439839423, 2011.

[20] A. Srivastava, A. Lanzi, J. T. Giffin, D. Balzarotti. Operating System Interface Obfuscation and the Revealing of Hidden Operations. 8th Conference on Detection of Intrusions and Malware and Vulnerability Assessment (DIMVA'11), Amsterdam, NL, July 2011, Springer-Verlag, ISBN: 978-3-642-22423-2, 2011.

[21] D. Gao, M. K. Reiter, and D. Song. Beyond Output Voting: Detecting Compromised Replicas Using HMM-Based Behavioral Distance. IEEE Transactions on Dependable and Secure Computing, Vol. 6, No. 2, 2009.

[22] S. Afrosa, A. Hamou-Lhadj, and M. Couture. An Improved Hidden Markov Model for Anomaly Detection Using Frequent Common Patterns. IEEE International Conference on Communications, Communication and Information Systems Security Symposium, Ottawa, ON, October 2011.

[23] A. Patcha and J.-M. Park. An Overview of Anomaly Detection Techniques: Existing Solutions and Latest Technological Trends. Elsevier, Computer Networks, Vol. 51, No. 12, 2007.

[24] W. Lu and A. A. Ghorbani. Network Anomaly Based on Wavelet Analysis. EURASIP Journal on Advances in Signal Processing, Vol. 2009, 2009.

[25] L. Ying, Z. Yan, and O. Yang-Jia. The Design and Implementation of Host-based Intrusion Detection System. The 3rd International Symposium on Intelligent Information Technology and Security Informatacis, Jinggangshan, CN, 2–4 April 2010, IEEE Computer Society, 2010.

[26] J. Gharajedaghi. Systems Thinking – Managing Chaos and Complexity. Butterworth-Heinemann, ISBN 0750671637, 1999.

[27] C4ISR Architecture Working Group. Levels of Information Systems Interoperability (LISI). 1998.

[28] G. Hallingstad and L. Dandurand. Cyber Defence Capability Framework. NATO C3 Agency, Document No. 3060, 2010.

[29] A. Hamou-Lhadj. Analysis of Redundant-Diverse Information Systems, and Proofs of Concept. Defence R&D Canada, DRDC Valcartier, DRDC Contract # W7701-001423, 2011.

[30] R. Khoury, A. Hamou-Lhadj, M. Couture, and R. Charpentier. Diversity through N-Version Programming: Current State, Challenges and Recommendations. Accepted for publication in the International Journal of Information Technology and Computer Science (IJITCS), 2012.