Defence Research and Development Canada
Recherche et développement pour la défense Canada

Cyber Anomaly Detection

A selected literature review

Etienne Martineau
DRDC Valcartier

Defence R&D Canada – Valcartier

Technical Memorandum
DRDC Valcartier TM 2011-417
June 2012

Canada

Cyber Anomaly Detection

A selected literature review

Etienne Martineau
DRDC Valcartier

Defence R&D Canada – Valcartier

Technical Memorandum
DRDC Valcartier TM 2011-417
June 2012

Principal Author

Original signed by Etienne Martineau

Etienne Martineau

Defence Scientist

Approved by

Original signed by Stéphane Paradis

Stéphane Paradis

Section Head/Intelligence & Information Section, DRDC Valcartier

Approved for release by

Original signed by Christian Carrier

Christian Carrier

Chief Scientist, DRDC Valcartier

Abstract

The cyber security domain is becoming a priority for the Canadian Forces and its allies. Potential threats presented by terrorism, espionage and even warfare are enough to raise security concerns. Conducted within the framework of cyber security, this work builds on projects 11hg and 11hk on maritime anomaly detection, by exploiting the abstract nature of anomaly detection. The aim of this document is to first present anomaly detection in general, and then to summarize recent publications on it in the cyber domain. The topics explored are program execution, network flow, payload inspection, and honey pots. Furthermore, papers on specific applications such as SCADA network, automotive systems, and network clusters are also part of the reviewed material. It has been found that while anomaly detection is not a solution to cyber threat on its own, it can be a valuable tool to detect zero-day attacks. A direct consequence of this observation is that cyber anomaly detection offers many research opportunities for detecting when systems are used outside their designed boundaries or in an abusive manner.

Résumé

Le domaine cyber devient une priorité pour les forces canadiennes et leurs alliés. La menace potentielle qu'il peut présenter à travers le terrorisme, l'espionnage et même la guerre est suffisante pour susciter l'inquiétude au niveau de la sécurité. Ce travail exploite les efforts de détection d'anomalies maritimes effectués dans les projets 11hk et 11hg en réutilisant l'aspect générique de la détection d'anomalies. L'objectif de ce document est de présenter les généralités de la détection d'anomalies en premier lieu, pour ensuite résumer les publications récentes sur le sujet dans le domaine cyber. Les sujets abordés sont : l'exécution de programmes, le débit des réseaux, l'inspection de paquets de données et les « honeypots ». De plus, les publications sur les applications spécifiques comme les réseaux SCADA, les systèmes pour véhicules et les clusters font aussi partie du matériel examiné. Il a été observé que même si la détection d'anomalies n'est pas d'elle-même une solution aux menaces cyber, elle peut être un outil de grande valeur pour la détection des attaques « zero-day ». Une conséquence directe de cette observation est que la détection de cyber anomalies offre plusieurs opportunités de recherche pour l'identification de systèmes qui sont utilisés en dehors de leurs limites conceptuelles ou d'une façon abusive.

This page intentionally left blank.

Executive summary

Cyber Anomaly Detection: A selected literature review

Etienne Martineau; DRDC Valcartier TM 2011-417; Defence R&D Canada – Valcartier; June 2012.

Introduction or background: The cyber security domain is becoming a priority for the Canadian Forces and its allies. Potential cyber threats presented by terrorism, espionage and even warfare are enough to raise security concerns. For example, the emergence of previously unknown cyber attacks that are technologically ahead of common defence. As a first step to address the problem, this work builds on projects 11hk and 11hg concerning maritime anomaly detection. It investigates how domain agnostic anomaly detection is used in recent cyber domain applications.

Results: This document presents anomaly detection in general, and then summarizes recent publications on it in the cyber domain. The topics explored are program execution, network flow, payload inspection, and honey pots. Moreover, papers on specific applications such as SCADA network, automotive system, and network cluster are also part of the reviewed material.

Significance: Most of the R&D activities on maritime anomaly detection conducted under projects 11hg and 11hk were focused on providing domain agnostic technologies that could be reused in other contexts. Driven by the objective of exploring other usages for these technologies, the work reported attempts to address a difficult problem in defence and security, i.e., cyber security.

Future plans: While anomaly detection is not a solution to cyber threat on its own, it can be a valuable tool for detecting previously unknown cyber attacks. Cyber anomaly detection offers many research opportunities for detecting when systems are used outside their designed boundaries or in an abusive manner. These opportunities will be further investigated and could possibly be addressed by the proof-of-concept prototypes already developed under projects 11 hg and 11hk.

Sommaire

Cyber Anomaly Detection: A selected literature review

Etienne Martineau ; DRDC Valcartier TM 2011-417 ; R & D pour la défense Canada – Valcartier; juin 2012.

Introduction ou contexte : Le domaine cyber devient une priorité pour les forces canadiennes et leurs alliés. La menace potentielle qu'il peut présenter à travers le terrorisme, l'espionnage et même la guerre est suffisante pour susciter l'inquiétude au niveau de la sécurité. Par exemple, l'apparition d'attaques cybernétiques jamais aperçues auparavant qui sont en avance sur les défenses couramment utilisées. À titre de première tentative pour adresser ce problème, ce travail tire profit des efforts effectués dans le projet 11hk et 11hg sur la détection d'anomalies maritimes. Il recherche comment la détection d'anomalies génériques est utilisée dans les applications récentes dans le domaine cyber.

Résultats : Ce document présente la détection d'anomalies en général, puis résume les publications récentes dans le domaine cyber. Les thèmes explorés sont l'exécution de programmes, le débit des réseaux, l'inspection de paquets de données, et des « honeypots ». Par ailleurs, des documents sur des applications spécifiques telles que les réseaux SCADA, les systèmes automobiles, et les clusters font également partie du matériel examiné.

Importance : La plupart des activités de R & D sur la détection d'anomalies maritimes menées sous les projets 11hg et 11hk ont été destinée à fournir des technologies génériques qui pourraient être réutilisés dans d'autres contextes. Poussé par l'objectif d'explorer d'autres utilisations pour ces technologies, ce travail est important car il adresse un « hard-problem » identifié dans la défense et la sécurité, à savoir, la cyber-sécurité.

Perspectives : Il a été constaté que bien que la détection d'anomalies n'est pas en soi une solution à la menace cybernétique, elle peut être un outil précieux pour détecter les cyber-attaques qui n'ont jamais été vues auparavant. Une conséquence directe de cette observation est que la détection des anomalies cyber offre de nombreuses possibilités de recherche afin de détecter quand les systèmes sont utilisés en dehors de leurs limites conceptuelles ou de manière abusive. Ces possibilités seront encore étudiées et pourraient éventuellement être adressées par les prototypes déjà développés sous les projets 11hg et 11hk.

Table of contents

This page intentionally left blank.

1 Introduction

The world we live in is more connected than ever. This growing trend of the last few years is motivated by the increased capabilities it provides. Computers, phones, tablets, music players and game consoles are just a sample of the entire set of devices that can connect to some sort of network. The new capabilities provided by network connections are not just meant to provide added value to standalone devices; sometimes the device is just a portal to a large universe of services provided by the networks. Examples are the World Wide Web, Internet Protocol(IP)-phone, IP-TV, Supervisory Control And Data Acquisition (SCADA) systems, and all possible cloud services (Gmail, GoogleDoc, etc.). As the list of services grows and the adoption of those services increases, the world slowly becomes dependent of this cyber domain. Society relies more and more on connectivity and as a consequence it is becoming an interesting exploitation opportunity for malevolent entities.

The threat is not new; hackers have been around for decades. However, as our lives increasingly move to the cyber domain, the range of opportunities increases accordingly. The need to protect network-enabled systems has never been so high. Malicious cyber activities are growing both in number and in complexity. Moreover, some threats are always one step ahead, taking advantages of undisclosed/undiscovered vulnerabilities in network systems. These so called *zero-day threats* can be exploited for long period of time before someone notices them and devises appropriate counter measures.

Anomaly detection has been sought to detect and help react rapidly to these threats. The reason why is simple: there is always a flaw somewhere. History has taught us the valuable lesson that even if the software/hardware/systems/networks are carefully developed, tested and patched, someone may find a way to exploit them [Zanero, 2006]. Anomaly detection is not a solution to cyber threats; it is a tool that could provide early threat warning and should not be considered as the main arsenal of a cyber defence strategy.

The topic of anomaly detection in the cyber domain is addressed in the next section. The goal of this document is to provide basic knowledge of the techniques and an overview of their usages in the recent open literature. To achieve this, anomaly detection techniques are presented briefly in the first part, while the second part deals with their usage in the cyber domain as mentioned in scientific papers.

2 Anomaly detection overview

This section provides a high-level introduction to anomaly detection. It focuses on the anomaly detection lifecycle and the most common techniques used. A more complete discussion of this topic can be found in [Chandola & al., 2009] and [Hodge et al., 2004]. The content presented here is domain agnostic, i.e., it can be applied in many fields and it is not specific to the cyber domain. Anomaly detection specific to the cyber domain is the subject of section 3.

2.1 How is anomaly detection performed?

There is no consensus on the definition of the term "anomaly," and that causes some confusion about what is normal or abnormal. In the literature, several terms are used: "outliers," "anomalies," "unusual," etc. One way to sum up all of these definitions could be as follows: a perspective from which an observation does not seems to belong to any previously known group. That definition is, of course, debatable. Anomaly detection refers to the problem of finding divergent values in a data set were one can observe a general pattern. Anomaly detection is far from being new; statisticians have been doing it formally since as early as the 19th century. Techniques for detecting anomaly are well known and have been used in many contexts over the years. The real challenge is to have access to data where a general normal pattern is distinguishable from unwanted behaviour and to find applicable anomaly detection techniques associated with this data.

Data can be separated in two categories: training data and working data. The former is used to build the basic awareness of the working environment where anomaly detection will be performed. This awareness will be used to define the normality patterns. In some cases, the data will be labelled as normal or abnormal giving the opportunity to assess the performance of the detection before it goes online. The working data is where anomaly detection is performed. This data can also be used to evolve the normalcy patterns if the patterns are subject to change over time.

Models and patterns of normalcy are extracted from the data. These models will be used to test the incoming data to see if it can be explained in part by them. These patterns may not always exist, but studies have shown that, most of the time, they do. By definition, a pattern is composed of recurring events that repeat in a predictable manner. Patterns can be, for example, a sequence of events, a statistical distribution, or a cluster of elements. Since models are low information representations of past data, they cannot (most of the time) describe exactly data from the past or the future. However, there is an upside to this: a model should not be too strict or precise. A good model will capture major trends while avoiding overfitting the training data. Also, patterns in the data may change overtime so it may be desirable to have evolving models or just to rebuild them once in a while. If the patterns are not subject to change over time, models are said to be static. Models can also be dynamic if the patterns evolve over time.

The actual anomaly detection is performed by testing new data against the model. Typically, outputs of an anomaly detection technique are scores or labels. Scores are often provided by statistical methods and give the opportunity to set a threshold on the number of anomalous events. Labels are the result of a classification more typical of machine learning approaches. Problems at

this stage are the number of false positives and the poor detection rate. The perfect case where all anomalies are detected without errors is rarely seen. On one hand, if the detection reports too few candidates, it may miss some anomalies. On the other hand, too many candidates dilute the real anomalies inside false positives providing useless results. The right anomaly detection and its configuration parameters technique must be chosen carefully to minimise these problems.

2.2 Techniques used

There are many methods for discovering patterns. The choice of a particular algorithm depends on a number of factors. The data format, performance requirements and the nature of the anomalies are some considerations that influence the choice of a particular method. Describing the techniques used to detect anomalies is a colossal task that is beyond the scope of this document. However, the main families of solutions, taken from [Hodge et al., 2004], are presented here in order to lay the groundwork for later sub-sections of this document where the practical applications of anomaly detection are presented.

2.2.1 Statistical methods

There are two types of statistical techniques: parametric and nonparametric. With parametric techniques, if the data correspond to a particular statistical model, anomalies can be detected rapidly and without supervision. With nonparametric techniques, no assumption is made about the underlying distribution of the data. Although more resources are required to develop them, these methods are effective for automated anomaly detection.

Statistical techniques are simple to implement, but their capability is limited to specific problems. Traffic volume is a good example of a variable where these techniques are effective because the anomalies are extreme values. In cases where anomalies are uniformly dispersed in the sample, these techniques are ineffective. Moreover, since it is difficult to define a threshold for separating abnormal values in a normal distribution, statistical techniques are likely to have a high level of false positives.

2.2.2 Neural networks

There is a great deal of literature on neural networks, and they come in many varieties. Overall, we can say that they generalize well to unseen patterns and are capable of learning complex class boundaries. After training, the neural network acts as a classifier. However, all data must be traversed several times before the network converges to an appropriate data model. Training and testing are also required in order to fine-tune the network and determine threshold settings before neural networks are ready to be used for classifying new data.

One of the biggest criticisms of neural networks is that the process is very obscure; they are often referred to as black boxes. The processing between the input and output neurons is not intelligible and cannot provide the operator with explanations or justifications. In addition, they are subject to overfitting: if the learning phase is too strict or targeted, then classification performance on data near a class boundary may drop.

2.2.3 Machine learning

Machine learning is not a technique but rather a field of research. It is a candidate of choice for anomaly detection. The main focus of the discipline is to automatically learn complex structures and make decisions based on the data. This focus is similar to that of an anomaly detection system.

Several documents mention that the use of machine learning is desirable, but little detail is provided. This reflects the diversity of problems in anomaly detection. Indeed, there is no single technique that meets all the requirements; restrictions typical of each situation can be very different. Here is a short list of popular techniques belonging to this field: decision trees, genetic programming, support vector machines, Bayesian networks, clustering, etc.

2.3 In the literature

Anomaly detection is not domain specific. As a consequence, there is plenty of literature on the topic in different domains. While specific applications of it outside the cyber domain are not much of interest in this document, there are some general surveys and introductions that go deeper than what was presented earlier in this section. For the reader interested in a wider view of the application of anomaly detection and a deeper look at the techniques, [Chandola & al., 2009] and [Hodge et al., 2004] are worth a look.

Work has also been done to review different aspects of anomaly detection in the cyber domain. A comprehensive survey of anomaly detection systems and hybrid intrusion detection systems of the recent past is provided in [Patcha & al, 2007]. They also discuss recent technological trends in anomaly detection and identify open problems and challenges in this area. While not specific to anomaly detection, the review of [Faysel & al, 2010] of intrusion detection systems and intrusion prevention systems is also based on this topic. It describes how the different techniques are applied in concrete computer systems. A survey of cyber attacks [Singh & al 2009] also mentions anomaly detection as a potential analytical approach to attack detection. In [Meza & al, 2009], cyber security is sought by advanced mathematics and statistics as a rich set of new and exciting research opportunities through, in particular, anomaly detection.

Beyond the conventional computer system or network, there is another application of cyber anomaly detection. Cyber attack on industrial control systems, also called SCADA systems, gained a lot of attention lately with the highly mediatised Stuxnet attack. Anomaly detection on these systems is discussed in [Dussel & al, 2009] and [Linda & al, 2011]. Also, less know specialized automotive bus systems can also be the target of cyber attacks. Alongside the benefits of an increased connectivity and functionality that come with modern vehicles come the increased exposure and vulnerability. This topic is explored in [Müter & al, 2011].

3 Uses of anomaly detection in the cyber domain

Following the general presentation of the field given in the previous section, we now turn to specific topics of anomaly detection in the cyber domain. These topics have been selected from recent publications in the open literature. The goal here is not to provide an exhaustive review of all the possible applications of anomaly detection but to provide an overview of the latest trends on the topic. For each reviewed paper, the targeted application is provided along with the anomaly detection method and an assessment of the performance.

3.1 Program execution

Using a high-level abstract definition, a program is a task list given to a processing unit. A task list defines exactly what a processing unit will do and leaves no place for uncertainty. However, in some cases, the program can be changed by an unauthorised entity to perform tasks that it was not meant for in the first place. These changes can be done by, for example, malicious code injection following a buffer over flow exploitation or a simple format string vulnerability. In any case, it is important to detect these changes to provide an early warning to the system administrator.

Monitoring the system calls that processes make is one way to collect data and perform anomaly detection. Finite automaton can be used for this task. A system designed for this is first trained on normal usage to register the automaton states and possible transitions. When put online, an anomaly is detected when an invalid transition is performed, i.e., a never seen before sequence of system calls. [Gao & al, 2004] propose a design space for this kind of approach to system calls anomaly detection. The result, a classification of automaton design, is used to highlight the costs and benefits of different nondeterministic finite automaton approaches. This classification will also be used to better combine different designs afterward. The design space is organized along three axes: 1) the information extracted from the process on each system call; 2) the granularity of the atomic units utilized in anomaly detection (single system call or variable-length system call sequences); and 3) the history of such atomic units (remembered by the anomaly detector as a stream during the monitoring phase). This framework enables the categorization of most previous approaches and help pinpoint new approaches that were not explored before. The authors make some recommendations on the design of automaton and evaluate possible ways and attacker could bypass or escape the anomaly detection process.
[Baah & al, 2006] presents a new machine-learning technique that performs anomaly detection on deployed software. Probes are first inserted in the original software to monitor branching. After this instrumented version is compiled and built, it is run upon a test suite. This training phase records the sequence of predicate values and their location in the code, i.e., class, method, and line number. These predicates are then clustered to form states that will be used to perform modeling with fully observable Markov model. An anomalous state is added to the model for unseen predicate transition. When an anomaly is detected, the developer can easily trace back the error and debug the software. Moreover, it is claimed that the software faults can be fixed before they cause critical failures in the system. The amount of developer's time used to debug the deployed software is also reduced. While this method was not designed to detect attacks on

software, one can use it to detect malicious code injection. In a case study, this method detected 50% of the anomalies before software failure.

Executable programs security through neural network anomaly detection has also been tried. In [Pan & al, 2009], an improved hybrid neural network made of radial basis function networks and self organizing map networks is used. In the process of profiling normal program behaviours, monitoring points inside a program are selected to extract its critical characteristics. Monitoring point states that will be fed to the networks include function call sequences, the context of program vulnerabilities, stack segment, heap segment and data segment. Function call sequences and the context of program vulnerabilities can be extracted through the static analysis of binary code. The characteristics of stack segment, heap segment and data segment can be acquired through the dynamic analysis of memory space during program execution. For performance evaluation, the authors have chosen intrusion detection datasets released by the University of New Mexico. Networks are first trained from this data set on normal execution, buffer overflow attack on the stack, buffer overflow attack on the heap, format string attack and memory reference/free attack. Networks are then put to the test using the Metasploit penetration testing software. Results show that this scheme detects 85% of the attacks with only 4% of false positives.

Rules can also be used to detect anomalous behaviour in a program. The method exposed in [Ming & al, 2011] proposes a scheme to seek evidence of ground truth to support rule-based anomaly detection. The approach is based on the analysis of the parameters flow in system calls. Sequences of system calls are intercepted, and parameters and return values are marked as tainted if they have been changed during calls. After the training run, rules are built from untainted parameters, based on the assumption that these patterns are static throughout the execution of the software. Results show that running the rules mining process on ground truth observations produces considerably fewer rules than when it is run on all observations. It also follows that the processing overhead is also reduced. However, comparative results on the actual anomaly detection performance are not provided. The use of system call arguments for anomaly detection is also the topic of this high-level presentation [Zanero, 2006].

The main drawback of the anomaly detection methods presented so far for program execution is the loss of performance. This concern, with evaluations and considerations, is mentioned in [Gao & al, 2004], [Pan & al, 2009] and [Ming & al, 2011]. The work in [Zhang & al, 2005] attempts to overcome this issue by proposing hardware-based anomaly detection systems. The advantage of being burned on a chip is that it is temper proof. It is also faster so one can perform more detailed anomaly detection and do it in real time, thus providing better security. The approach is based on anomalous path checking. During the training phase, the sequences of jump instructions on the execution path are recorded. From that, all the possible jump sequences of length N are extracted and kept as the normal execution sequences that the software can perform. During the detection phase, a sliding window covering the sequence of the last N jump instructions is tested again the possible normal sequences; if the sequence is not found, an anomaly is detected. As the number N gets bigger, so does the security and unfortunately the false positive rate. It is claimed that all the conventional buffer overflows and stack smashing attacks can be detected. Also the detection scheme with N=5 can detect random branch diversions with a success rate above 97%.

3.2 Network behaviour

Changes in network behaviour can sometime be the result of malicious activities. Abnormal behaviours can take many forms: port scan, Transmission Control Protocol(TCP) SYNchronization packet (SYN) flood, egg download, and spam to name a few. Since these behaviours standout from normal usage, it is a good opportunity to perform anomaly detection. As for anomalous program executions, it is important to detect unexpected changes to provide an early warning to system administrators. Descriptions of such anomaly detection attempts are presented next.

To detect the presence and prevent further propagation of fast scanning worms, it is proposed in [Whyte & al, 2005] to declare anomalous all connections that are not correlated with an address resolution query. Rules to classify normal connections ensure that the destination IP address of an outgoing TCP connection or User Datagram Protocol (UDP) datagram can be attributed to either a DNS query, an HyperText Transfer Protocol(HTTP) packet with an embedded address, or an entry in a user defined white list. These rules are assigned a time to live equal to the one in the Domain Name Server(DNS) queries; those from the white list that are permanent. The system will inspect TCP packets with the SYN only flag and try to find a matching rule; if no rule is found, an alert is raised. While this method can block unauthorised outbound connections, it cannot block incoming connections. Also, connections inside the same network that rely on Address Resolution Protocol(ARP) cannot be blocked. Another drawback is the necessity to rebuild the white list every time a new application that uses direct IP connection is used on the network (this is much like changing rules in a firewall). Some evaluation of the method is provided in the document but it does not contain the actual detection rate and the associated false positive rate.

In contrast to the work in [Whyte & al, 2005], ARP based network traffic anomaly detection is the subject of [Yasami & al, 2007]. This method can detect DoS attacks, virus, and worms evolving in a local network. However, it cannot handle ARP spoofing since it does not alter the network flow. ARP traffic is modeled for every node as a hidden Markov model. States of the model correspond to the nodes in the network and transitions are triggered by ARP requests. Anomaly scores are computed on ARP sequences based on the probability of state transition and the time spent in each state. So nodes that perform many requests in a short period of time or to inexistent node will trigger an alert. The system is trained on normal network traffic before being put to work. Experimental results on two months of network traffic at a university campus have claimed a detection performance above 90%. However, the paper does not mention the false positive rate.

Similarly, network flow anomaly detection in backbone is the topic of [Dewaele & al, 2007]. The proposed anomaly detection method targets DDoS, flooding, flash crowds, and worm outbreaks through single point measurement. Packets are analyzed within sliding time-windows using a statistical method. They are first categorised using random hash functions before the attributes (time stamps, IPs and ports) from each category are used to create time series. Note that the payload itself is not analysed. Time series are then described using Gamma laws. An anomaly is detected when a change in the parameter of the laws is detected among categories using the Mahalanobis distance. One of the main advantages of this method is that it does need a training phase. Also, it can detect long-lasting/low-level intensity attacks as well as the more aggressive and high throughput short-lived ones. The method was tested on recorded traffic with results claimed to be satisfactory although no numbers are given on the false positive

and false negative rates. On the performance side, this scheme can handle 15 minutes of backbone traffic in one minute on a desktop computer.

The work in [Huang & al, 2007] also proposes to detect the same kind of anomaly, i.e., traffic volume anomaly. The proposed scheme used some nodes in the network that possess a sensor that registers time series based on network traffic (TCP connections, DNS requests, etc.). These time series are filtered locally and when the node judges it necessary, it sends them to a coordinator node where the anomaly detection will take place. The coordinator then performs principal component analysis on the aggregation of the data received from the sensors and uses the minor components (lowest eigenvalues) to detect anomalies. These components represent the abnormal traffic subspace and an alert is raised when the traffic volume in this subspace is over a predefined threshold. Test on a simulator (based on real traffic snap shots and injected anomalies) yield, approximately, a 4% missed detection rate and a 6% false alarm rate. It is also interesting to note that the filtering is claimed to provide an upper bound on the false alarm rate.

[Lim & al, 2009] also proposes distributed sensors with a coordinator. This conceptual view is not based on any particular anomaly detection technique. However, the detection process proposes to apply one or more anomaly detection techniques in a hierarchical manner. Anomaly detection is first done at the node level before being escalated to the coordinator. This design prevents potential performance bottle neck in the coordinator. The first stage of the hierarchical detection system requires only a low level of analysis effort. It only observes packet distribution. Further stages are then loaded whenever an adverse event is assumed in first stage. These further stages detect only anomalies based on the information about the adverse event assumed in the first stage. Being only in the conceptual phase, this scheme has not been tested by the authors.

Most anomaly detection papers do not address the aspect of mitigation. The method exposed in [Sun & al, 2010] is a detection-mitigation algorithm. The goal is to regulate the data flow over the network. The first assumption is that legitimate traffic adapts to the congestion state much like the behaviour of the TCP protocol. The second is that abnormal traffic aggregate around one IP address. An anomaly is detected when the ratio of the aggregate traffic around one IP address over the whole traffic surpass a certain threshold. In that situation, the system will start dropping a small percentage of packets from/to the concerned IP address. Legitimate traffic will adjust while abnormal traffic will not, making it more detectable. Packet drop on abnormal traffic will increased gradually giving more bandwidth to legitimate traffic. Tests were conducted on some university networks where the anomaly detection system resided on the router. Attackers were simulated with different kind of denial of services attack. While no numbers are given, it is claimed that the system succeeded in filtering out malicious traffic while delivering the legitimate one. It is also mentioned that it has a high flexibility and a low rate of false detections.

3.3 Payload inspection

The flow of packets is an aspect on which one can perform anomaly detection to try to catch unwanted usages. However, as mention in [Whyte & al, 2005], some threats do not alter the flow of to the network and thus cannot be detected by methods presented in the previous section. To mitigate these potential threats, one has to inspect the payload of each communication. Anomaly

detection in payload is a form of deep packet inspection where one searches for abnormal data streams instead of known signatures.

To achieve this, Support Vector Machines (SVM) are used in [Perdisci & al, 2006] to classify HTTP packets. The method is inspired by the success of text classifiers based on the same technique. By using a sliding window of fixed length, the pair made of the first and last byte of the window is used as a features vector. By changing the size of the sliding window, a representation of the payload in different feature spaces is created. A feature clustering algorithm is then used to reduce the dimensionality in which it is represented. Finally, detection accuracy and hardness of evasion are obtained by constructing a combination of multiple one-class SVM classifiers that work on these different feature spaces. One advantage of this method is that by assuming that the training dataset contains only normal requests, the rejection rate of the SVM can be interpreted as a desired false positive rate. Experimental results reported in the paper support this assumption. With a desired false positive of 0.01%, the actual false positive is 0.018% with a detection rate of 96%. The topic of the processing is said to be left for future work but the authors mention that the performance does not appear to be an issue. In later work ([Perdisci & al, 2009]) the authors mention that the performance is less than PAYL [Wang & al, 2004] which is not fast enough to process a large amount of traffic [Thorat & al, 2008]. Also, a more complete evaluation is performed: tests were done with the 1999 Defense Advanced Research Projects Agency (DARPA) Intrusion Detection System (IDS) data set [Lippmann & al, 2000] in combination with different mimicry attacks.

In [Dussel & al, 2009] the authors are using the TCP packets re-assembly software BRO for the purpose of anomaly detection. Payload byte sequences are mapped into a feature space vector where features can be the payload length, the presence of a particular string or the frequency of a term. Using predefined training points (testing features) a model is build using normal sequences of network traffic. Anomaly detection is performed using a new sequence of data that is tested against features in the model using a similarity measure. The similarity measure between byte sequences is determined by the computation of the pair wise distance between vectorial feature representations. The resulting vector is declared anomalous if it is too far from the centroid given by the training data. With detection rates of 88%-92% at a false positive level of 0.2%, the method has been proven to be useful for the detection of unknown attacks in network traffic. This performance is function of the number of testing features. A high number of features increases the accuracy but also reduces the network throughput.

Single byte frequency distribution is used in [Thorat & al, 2008] to detect payload anomalies on TCP port 21 and 80, i.e., File Transfert Protocol (FTP) and HTTP. The paper discusses an improvement of the PAYL [Wang & al, 2004] software which is considered as a complete system for payload-based anomaly detection but lacks the speed to be used on high bandwidth networks. To improve the throughput of PAYL, it is suggested to reduce the amount of payload scanned. The proposed method used a sliding window to scan and split the payload in content blocks using Rabin fingerprinting. However, instead of scanning the entire payload as in PAYL, the system has a stopping criterion based on the number of content blocks generated by the sliding window. Distributions are created on aggregation of content blocks of similar length for each network port. To perform anomaly detection, incoming payload byte frequency distributions are tested against the stored model from the training phase using a simplified Mahalanobis distance. This scheme has been tested using the 1999 DARPA IDS data set [Lippmann & al, 2000] and has been found

to detect more than 96% of the attacks while using less than 62% of packet payload on average. The false positive rate was below 0.18%.

To lower the false positive rate, collective anomaly detection is used in [Boggs & al, 2011] to detect zero-day attacks. The system inspects user submitted content inside HTTP GET requests to create a model of normal requests based on normal traffic sampling. It is not performing the anomaly detection on its own but leverages currently existing softwares. The paper proposes to normalise GET requests before using a sliding window that split it in strings of 5 bytes. These strings are then hashed to provide the position in a Bloom filter (space-efficient probabilistic data structure) where the occurrences of the strings are recorded. These filters are the normalcy models that are shared across the nodes to lower the false positive rate. If a node in the network detects an anomaly, i.e., a string not in the model, it asks the other nodes for validation before raising an alert. In case of a false positive validated by the operator, the faulty string is added to the filter to prevent further false alerts across the network. The authors claim (with 3 servers) a 0.03% false positive rate but do not provide the associated detection rate. The speed does not seem to be an issue since the detection only scans GET requests. One advantage of this method is that the normalisation of requests preserves the privacy of legitimate requests and thus protects the users. Also, the system can be adapted to perform peers validation with other anomaly detection schemes.

3.4 Honeypot

Honeypots are systems that appear to be normal in a network, but are actually traps used to detect and analyse potential threats. They are isolated and monitored with a strictly defined normal behaviour, which make them excellent candidates for anomaly detection. Since the normal behaviour is predefined, unplanned activity on honeypots are always anomalies. This section describes specific implementation of anomaly detection with honeypots.

Instead of using only one honeypot, virtual machines are used in [Dagon & al, 2004] to cover large IP space with decoys using only one machine. Virtual machines are loaded with different systems to increase the chance to catch worm attacks. The example provided includes Linux and Windows systems with different patch levels. Each honeypot monitors three different kinds of event: disk accesses, network activities and memory events. Events are then correlated with other observed events using logistic regression and other system statistics. If a common pattern emerges among virtual machines, this can indicate the presence of a worm or other automated attacks. If a machine initiates a network event, it is automatically reset to the original state to prevent worm propagation. Using the collected events and the types of operating systems, the honeypot reports an explanation of worm activation, and not merely the presence of a worm. It is claimed that this system can provide an effective way to detect worms early. The paper mentions the difficulties in trying to evaluate the system performance since activities on honeypots are always unwanted. Because the system was not tested with real worms, the detection performance and the false positive rate are not provided.

In [Anagnostakis & al, 2005] honeypots are used to filter the results of anomaly detection algorithms on network traffic. The proposed system has three phases. First, a signature-based filter is applied. Anomaly detection is performed in second, in order to split the legitimate traffic from the suspicious one. Finally, the suspicious traffic is analysed in a honeypot while the

legitimate traffic is processed as usual. The honeypot is an instrumented version of the original server. As a consequence, normal traffic going through it is processed normally (except for the delays introduced by the instrumentation). However, threatening requests will be detected, connections will be reset, and changes in the software will be rolled back. This method has the merit of reducing the false positive rate while keeping a good server throughput. Also, the attack mechanism can be analysed in the honeypot to adapt the filters in the previous two phases. The main drawback of this approach is the need to build an instrumented version of the software that can detect attacks. This can be done when one has access to the source code (Apache in the paper) but cannot be used in closed source software. Detection rate and false positive being a function of the anomaly detection in phase two, no specific performance numbers are provided.

3.5 Domain specific

This section explores domains of cyber anomaly detection that are less common. So far, the presented material was based on classical network and computer security. However, anomaly detection can be applied to specific cyber domains. Less common applications are presented below.

Anomaly detection of in-vehicle networks is the topic of [Müter & al, 2011]. Modern vehicles possess many control systems that interact with each other's over specialized automotive bus systems. However, new communication media hooked to these buses (like GSM, 802.11, or Bluetooth) increase their exposure and vulnerability. Among exposed features of the vehicle system, there are controls for speed, air bags, breaks and windows to name a few. The paper explains how the attack can take place and potential detection methods using no prior information (blackbox) or with the help of some knowledge of the system (graybox). The focus is placed on entropy-based anomaly detection since the communication over automotive bus systems has near constant low information contain. Information theoretic measures are used to detect, for example, flooding of security critical components (Chassis- or Powertrain-Domain) and speed signal spoofing. Results provided show that automotive systems are good candidates for anomaly detection. While no numbers are provided the low level of randomness of these networks makes them easy to model, and consequently, anomalies are easier to detect.

When an anomaly occurs, it means that something wrong is going on. The topic of [Gu & al, 2009] is anomaly prediction, i.e., detecting future anomalies to prevent them. The area of interest of this paper is the prevention of bottle necks in cluster systems. Since loss of performance can be caused by various reasons (e.g., insufficient processing resources, insufficient memory, and memory leak bug) the authors propose to classify the state of the cluster using a naive Bayesian classifier where the inputs are discretized system and component metrics of the host. To predict future anomalous states, these metrics are themselves modelled using Markov models. After a training phase on real data, metrics are predicted few discrete steps in advance before being handled by the classifier. The result of anomaly detection provide which host in the cluster is about to fail, the symptoms, and in how much time. Experiments show that this approach efficiently predicts and diagnoses several bottleneck anomalies with high accuracy while imposing low overhead to the cluster system. However, the false positive rate is quite high, between 10% and 20%.

Anomaly detection in the network flow of industrial control systems is explained in [Linda & al, 2011], in which the authors propose a learning algorithm for a fuzzy logic based anomaly detection system specifically developed for the constrained resources of TOFINO, an embedded network security cyber sensor. The device has been developed for pre-emptive threat detection, termination and reporting, specifically tailored for the needs of SCADA systems. Using the normal network communication behavioural patterns, a fuzzy rule base is constructed with the help of the online version of the nearest neighbour clustering algorithm. This scheme focuses on modeling the time series behaviour of different features in the packet stream. These features are (or relate to) for example: Number of IP addresses, protocols, flag codes, window sizes and data length. On a simulated intrusion attempt including ARP pings, SYN stealth scans, port scanning, open port identification, the system maintained a 0% false positive rate and a 0.9% average false negative rate. The reported processing speed of the device is over 5000 packets per second.

4 Conclusion

This document presents anomaly detection techniques and their applications in the cyber domain. The way these techniques can be used is limitless. The increasing number of new devices, software, hardware and systems that are networked provide new capabilities for systems but are also opportunities for malicious cyber activities. Potential unknown threats may exist and those components provide parameters that could be monitored and modelled for normalcy, increasing the potential applications of anomaly detection to achieve better security.

However, anomaly detection in general suffers from a major drawback: the high false positive rate. A cyber anomaly detection system must be carefully chosen/designed to be efficient at detecting threats without drowning operators with false alerts. The overhead of these systems is also a concern; they can easily become resource hug and cripple performances. Cyber anomaly detection is not a cyber security solution; it is not the answer to a known problem. However, its goal is to raise alerts when potential new problems appear. It is a "sense" component that increases situational awareness and by doing so, gives the opportunity for one to react more swiftly to cyber threats.

Cyber anomaly detection offers many research opportunities. Finding monitoring points to model normalcy efficiently in a cyber system requires expertise in the system in question. For example, detecting anomaly in software execution require different knowledge than detecting anomalies in a network flow. So, to be relevant and have a major impact, research in this field must be well targeted. It must look a developing in depth knowledge of emergent cyber systems that present valuable exploitation opportunities to foreign entities. This knowledge will then be used to develop anomaly detection mechanisms that will enable future systems to detect when they are used outside their designed boundaries or in an abusive manner.

References

[Anagnostakis & al, 2005], Anagnostakis K. G., Sidiroglou S., Akritidis P., Xinidis K., Markatos E., Keromytis A. D., Detecting targeted attacks using shadow honeypots. In Proceedings of the 14th conference on USENIX Security Symposium Vol. 14. 2005, pp 9-9.

[Baah & al, 2006], Baah G. K., Gray A., Harrold M. J., On-line anomaly detection of deployed software: a statistical machine learning approach. SOQUA 2006. 2006, pp 70-77.

[Boggs & al, 2011], Boggs N., Hiremagalore S., Stavrou A., Stolfo S. J., Cross-domain Collaborative Anomaly Detection: So Far Yet So Close. In Proceedings of the 14th International Symposium on Recent Advances in Intrusion Detection. 2011.

[Chandola & al., 2009], Chandola V. , Banerjee A., and Kumar, V., Anomaly detection: A survey, ACM Computing Surveys (CSUR), v.41 n.3, pp. 1-58, July 2009.

[Dagon & al, 2004], Dagon D., Qin X., Gu G., Grizzard J., Levine J., Owen H., HoneyStat: Local Worm Detection Using Honeypots, In Proceedings of the 7 th International Symposium on Recent Advances in Intrusion Detection. 2004.

[Dewaele & al, 2007], Dewaele G., Fukuda K., Borgnat P., Abry P., Cho K., Extracting hidden anomalies using sketch and non Gaussian multiresolution statistical detection procedures. In Proceedings of the 2007 workshop on Large scale attack defense. 2007.

[Dussel & al, 2009], Dussel P., Gehl C., Laskov P., Buber J., Stormann C., Kastner J., Cyber-critical infrastructure protection using real-time payload-based anomaly detection. In Proceedings of the 4th international conference on Critical information infrastructures security. 2009 pp 85-97.

[Faysel & al, 2010], Faysel M. A., Haque S. S., Towards Cyber Defense: Research in Intrusion Detection and Intrusion Prevention Systems. International Journal of Computer Science and Network Security, VOL.10 No.7, 2010.

[Gao & al, 2004], Gao D. ,Reiter M. K., Song D. X., On Gray-Box Program Tracking for Anomaly Detection, In Proceedings of USENIX Security Symposium. 2004, pp 103-118.

[Gu & al, 2009], Gu X., Wang H., Online Anomaly Prediction for Robust Cluster Systems. In Proceedings of the 2009 IEEE International Conference on Data Engineerin. 2009.

[Hodge et al., 2004], Hodge V., and Austin, J., A Survey of Outlier Detection Methodologies, Artificial Intelligence Review, v.22 n.2, pp. 85-126, October 2004

[Huang & al, 2007], Huang L., Nguyen X. L., Garofalakis M., Jordan M., Joseph A.D., Taft N. In-network PCA and anomaly detection. In NIPS. 2007.

[Lim & al, 2009], Lim S. Y., Jones A, An Anomaly-based Intrusion Detection Architecture to Secure Wireless Networks. 2009.

[Linda & al, 2011], Linda O., Manic M., Vollmer T., Wright J., Fuzzy Logic Based Anomaly Detection for Embedded Network Security Cyber Sensor. In Proceedings of the Symposium on Computational Intelligence in Cyber Security (Preprint). 2011.

[Lippmann & al, 2000] Lippmann R., Haines J. W., Fried D. J., Korba J., Das K., The 1999 DARPA off-line intrusion detection evaluation, Computer Networks, Volume 34, Issue 4, Pages 579-595, ISSN pp 1389-1286, October 2000.

[Meza & al, 2009], Meza J., Campbell S., Bailey D., Mathematical and Statistical Opportunities in Cyber Security. ArXiv e-prints. 2009.

[Ming & al, 2011], Ming J., Zhang H., Gao D., Towards Ground Truthing Observations in Gray-Box Anomaly Detection. In Proceedings of the 5th international conference on network and system security (Preprint). 2011.

[Müter & al, 2011], Müter M., Asaj N., Entropy-Based Anomaly Detection for In-Vehicle Networks. In Proceedings of the IEEE Intelligent Vehicles Symposium. 2011.

[Pan & al, 2009], Pan W., Li W., Zhao W., A Novel Anomaly Detection Approach for Executable Program Security. In Proceedings of the 2009 International Conference on Multimedia Information Networking and Security. 2009.

[Patcha & al, 2007], Patcha, A. & Park, J. An overview of anomaly detection techniques: Existing solutions and latest technological trends. Computer Networks 51. 2007, pp 3448-3470.

[Perdisci & al, 2006], Perdisci R., Gu G., Lee W., Using an Ensemble of One-Class SVM Classifiers to Harden Payload-based Anomaly Detection Systems. In Proceedings of the Sixth International Conference on Data Mining. 2006.

[Perdisci & al, 2009], Perdisci R., Ariu D., Fogla P., Giacinto G., Lee W., McPAD: A multiple classifier system for accurate payload-based anomaly detection. Computer Networks: The International Journal of Computer and Telecommunications Networking vol 53 iss 6. 2009 pp 854-881.

[Singh & al 2009], Singh S., Silakari S., A Survey of Cyber Attack Detection Systems. International Journal of Computer Science and Network Security, VOL.9 No.5, May 2009.

[Sun & al, 2010], Sun Z., Gong J., Anomaly Traffic Detection Model Based on Dynamic Aggregation, Third International Symposium on Electronic Commerce and Security, 2010, pp. 46-50.

[Thorat & al, 2008], Thorat, S., Khandelwal, A., Bruhadeshwar, B. & Kishore, K. Payload Content based Network Anomaly Detection. 2008 First International Conference on the Applications of Digital Information and Web Technologies. 2008. pp 127-132.

[Wang & al, 2004] Wang K., Stolfo S., Anomalous payload-based network intrusion detection. In Recent Advances in Intrusion Detection, RAID 2004, pp 203–222, September 2004.

[Whyte & al, 2005], Whyte D., Kranakis E., Oorschot P. C. V., DNS-based Detection of Scanning Worms in an Enterprise Network. In Proceedings of the 12TH annual network and distributed system security symposium. 2005, pp 181-195.

[Yasami & al, 2007], Yasami Y., Farahmand M., Zargari V., An ARP-based Anomaly Detection Algorithm Using Hidden Markov Model in Enterprise Networks. In Proceedings of the Second International Conference on Systems and Networks Communications. 2007.

[Zanero, 2006], Zanero S., Anomaly detection through system call argument analysis. Black Hat briefings. 2006.

[Zhang & al, 2005], Zhang T., Zhuang X., Pande P., Lee W., Anomalous path detection with hardware support. In Proceedings of the 2005 international conference on Compilers, architectures and synthesis for embedded systems. 2005, pp 43-54.

List of symbols/abbreviations/acronyms/initialisms

ARP	Address Resolution Protocol
DARPA	Defense Advanced Research Projects Agency
DDoS	Distributed Denial of Service
DND	Department of National Defence
DoS	Denial of Service
DRDC	Defence Research & Development Canada
DNS	Domain Name Server
FTP	File Transfert Protocol
GSM	Global System for Mobile Communications
HTTP	HyperText Transfer Protocol
IDS	Intrusion Detection System
IP	Internet Protocol
R&D	Research & Development
SCADA	Supervisory Control And Data Acquisition
SVM	Support Vector Machines
SYN	Transmission control protocol SYNchronization packet
TCP	Transmission Control Protocol
UDP	User datagram protocol

This page intentionally left blank.

DOCUMENT CONTROL DATA

(Security classification of title, body of abstract and indexing annotation must be entered when the overall document is classified)

1. ORIGINATOR (The name and address of the organization preparing the document. Organizations for whom the document was prepared, e.g. Centre sponsoring a contractor's report, or tasking agency, are entered in section 8.)

Defence R&D Canada – Valcartier
2459 Pie-XI Blvd North
Quebec (Quebec)
G3J 1X5 Canada

2. SECURITY CLASSIFICATION (Overall security classification of the document including special warning terms if applicable.)

UNCLASSIFIED
(NON-CONTROLLED GOODS)
DMC A
REVIEW: GCEC April 2011

3. TITLE (The complete document title as indicated on the title page. Its classification should be indicated by the appropriate abbreviation (S, C or U) in parentheses after the title.)

Cyber Anomaly Detection: A selected literature review

4. AUTHORS (last name, followed by initials – ranks, titles, etc. not to be used)

Martineau, E.

5. DATE OF PUBLICATION (Month and year of publication of document.)	6a. NO. OF PAGES (Total containing information, including Annexes, Appendices, etc.)	6b. NO. OF REFS (Total cited in document.)
June 2012	32	30

7. DESCRIPTIVE NOTES (The category of the document, e.g. technical report, technical note or memorandum. If appropriate, enter the type of report, e.g. interim, progress, summary, annual or final. Give the inclusive dates when a specific reporting period is covered.)

Technical Memorandum

8. SPONSORING ACTIVITY (The name of the department project office or laboratory sponsoring the research and development – include address.)

Defence R&D Canada – Valcartier
2459 Pie-XI Blvd North
Quebec (Quebec)
G3J 1X5 Canada

9a. PROJECT OR GRANT NO. (If appropriate, the applicable research and development project or grant number under which the document was written. Please specify whether project or grant.)	9b. CONTRACT NO. (If appropriate, the applicable number under which the document was written.)

10a. ORIGINATOR'S DOCUMENT NUMBER (The official document number by which the document is identified by the originating activity. This number must be unique to this document.)	10b. OTHER DOCUMENT NO(s). (Any other numbers which may be assigned this document either by the originator or by the sponsor.)
DRDC Valcartier TM 2011-417	

11. DOCUMENT AVAILABILITY (Any limitations on further dissemination of the document, other than those imposed by security classification.)

Unlimited

12. DOCUMENT ANNOUNCEMENT (Any limitation to the bibliographic announcement of this document. This will normally correspond to the Document Availability (11). However, where further distribution (beyond the audience specified in (11) is possible, a wider announcement audience may be selected.))

Unlimited

13. ABSTRACT (A brief and factual summary of the document. It may also appear elsewhere in the body of the document itself. It is highly desirable that the abstract of classified documents be unclassified. Each paragraph of the abstract shall begin with an indication of the security classification of the information in the paragraph (unless the document itself is unclassified) represented as (S), (C), (R), or (U). It is not necessary to include here abstracts in both official languages unless the text is bilingual.)

The cyber security domain is becoming a priority for the Canadian Forces and its allies. Potential threats presented by terrorism, espionage and even warfare are enough to raise security concerns. Conducted within the framework of cyber security, this work builds on projects 11hg and 11hk on maritime anomaly detection, by exploiting the abstract nature of anomaly detection. The aim of this document is to first present anomaly detection in general, and then to summarize recent publications on it in the cyber domain. The topics explored are program execution, network flow, payload inspection, and honey pots. Furthermore, papers on specific applications such as SCADA network, automotive systems, and network clusters are also part of the reviewed material. It has been found that while anomaly detection is not a solution to cyber threat on its own, it can be a valuable tool to detect zero-day attacks. A direct consequence of this observation is that cyber anomaly detection offers many research opportunities for detecting when systems are used outside their designed boundaries or in an abusive manner.

Le domaine cyber devient une priorité pour les forces canadiennes et leurs alliés. La menace potentielle qu'il peut présenter à travers le terrorisme, l'espionnage et même la guerre est suffisante pour susciter l'inquiétude au niveau de la sécurité. Ce travail exploite les efforts de détection d'anomalies maritimes effectués dans les projets 11hk et 11hg en réutilisant l'aspect générique de la détection d'anomalies. L'objectif de ce document est de présenter les généralités de la détection d'anomalies en premier lieu, pour ensuite résumer les publications récentes sur le sujet dans le domaine cyber. Les sujets abordés sont : l'exécution de programmes, le débit des réseaux, l'inspection de paquets de données et les « honeypots ». De plus, les publications sur les applications spécifiques comme les réseaux SCADA, les systèmes pour véhicules et les clusters font aussi partie du matériel examiné. Il a été observé que même si la détection d'anomalies n'est pas d'elle-même une solution aux menaces cyber, elle peut être un outil de grande valeur pour la détection des attaques « zero-day ». Une conséquence directe de cette observation est que la détection de cyber anomalies offre plusieurs opportunités de recherche pour l'identification de systèmes qui sont utilisés en dehors de leurs limites conceptuelles ou d'une façon abusive.

14. KEYWORDS, DESCRIPTORS or IDENTIFIERS (Technically meaningful terms or short phrases that characterize a document and could be helpful in cataloguing the document. They should be selected so that no security classification is required. Identifiers, such as equipment model designation, trade name, military project code name, geographic location may also be included. If possible keywords should be selected from a published thesaurus, e.g. Thesaurus of Engineering and Scientific Terms (TEST) and that thesaurus identified. If it is not possible to select indexing terms which are Unclassified, the classification of each should be indicated as with the title.)

Anomaly detection, Cyber security

Defence R&D Canada

Canada's Leader in Defence
and National Security
Science and Technology

R & D pour la défense Canada

Chef de file au Canada en matière
de science et de technologie pour
la défense et la sécurité nationale

www.drdc-rddc.gc.ca