Information Fusion 11 (2010) 21–34

Contents lists available at ScienceDirect

Information Fusion

journal homepage: www.elsevier.com/locate/inffus

Information fusion for anomaly detection with the dendritic cell algorithm

Julie Greensmith *, Uwe Aickelin, Gianni Tedesco

School of Computer Science, University of Nottingham, Nottingham NG8 1BB, UK

ARTICLE INFO

Article history:
Received 30 January 2007
Received in revised form 21 September 2007
Accepted 1 April 2009
Available online 23 April 2009

Keywords:
Dendritic cells
Data fusion
Immune system
Anomaly detection
Port scans

ABSTRACT

Dendritic cells are antigen presenting cells that provide a vital link between the innate and adaptive immune system, providing the initial detection of pathogenic invaders. Research into this family of cells has revealed that they perform information fusion which directs immune responses. We have derived a dendritic cell algorithm based on the functionality of these cells, by modelling the biological signals and differentiation pathways to build a control mechanism for an artificial immune system. We present algorithmic details in addition to experimental results, when the algorithm was applied to anomaly detection for the detection of port scans. The results show the dendritic cell algorithm is successful at detecting port scans.

1. Introduction

Dendritic cells (DCs) are natural anomaly detectors. In this paper we present a dendritic cell algorithm (DCA) approach to information fusion, combining key elements of immunological theory with the engineering principles of data fusion. In the human immune system, DCs have the power to suppress or activate the immune system by correlation of signals representing their environment, combined with locality markers in the form of antigens. Antigens are proteins in structure and are any protein to which the immune system can potentially respond. These cells are responsible for the detection of pathogens in the human body through the correlation of information (in the form of molecular signals) within the environment. By using an abstraction of DC behaviour, similar detection properties are shown, resulting in an algorithm capable of performing anomaly detection. The resultant algorithm uses a set of weights derived or the processing of input signals from actual immunological data, generated through an interdisciplinary collaboration with immunologists [32].

DCs in particular are suitable as inspiration for intrusion detection for two reasons. Firstly, DCs themselves perform an intrusion detection role within the human immune system. Secondly, DCs perform their function with low rates of false positives and high rates of true positives – properties essential to any anomaly detection technique. In essence, DCs are multi-sensor data fusion agents through processing environmental molecular signals. This makes them ideal inspiration for the development of a data fusion algorithm.

The DCA was introduced in 2005 [8] and has demonstrated potential as a classifier for static machine learning data [8], as a simple port scan detector under experimental conditions [10] and in real time [9]. Our results show that the DCA can successfully detect anomalous processes forming a port scan attack. The DCA is inspired by the human immune system and is termed an *artificial immune system* (AIS). While the majority of AIS algorithms do not perform data fusion, idiotypic network models are used for the purpose of robotic control [12]. Although belonging to the field of artificial immune systems, the DCA differs from other immune inspired anomaly detection algorithms in a number of significant ways:

- The algorithm is based on cutting edge experimental immunology.
- DCs combine multiple signals to assess the current context of their environment.
- Asynchronously DCs sample another data-stream (antigen) to be combined with the fused signals.
- The correlation between context and antigen leads to the detection of anomalies.
- Unlike other anomaly detection algorithms, there is no pattern matching based on string similarity metrics.

The aims of this paper are threefold: to model artificial DCs drawing inspiration from the DCs of the human immune system; to present a resultant algorithm through a formalised description; and to apply the algorithm to an example anomaly detection

* Corresponding author. Tel.: +44 115 9514247.
E-mail addresses: jqg@cs.nott.ac.uk (J. Greensmith), uxa@cs.nott.ac.uk (U. Aickelin), gxt@cs.nott.ac.uk (G. Tedesco).

1566-2535/$ - see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.inffus.2009.04.006

problem. As this algorithm is a novel algorithm, it is not yet fully characterised. As a result, fine grained analysis of the selection of weights and comparison to other standard techniques are not discussed in this paper. Please refer to [11] for further experiments.

In this paper The DCA is applied to the detection of a port scan, which forms a convenient small-scale computer security problem. Section 2 contains relevant background information regarding the problem of port scans and current scanning detection techniques. Section 3 presents the biological inspiration of the DCA, a summary of relevant developments in immunology, and rudimentary DC biology. This is followed by Sections 4 and 5, describing the abstraction process, a formalised description of the DCA and its implementation as an anomaly detector. This is followed by experimentation with its application as a port scan detector. Section 6 includes a sensitivity analysis of a selection of parameters. The paper concludes with a discussion of the results of the port scan investigation and suggestions for future work.

2. Anomaly detection and port scanning

One notable application area of multi-sensor data fusion is anomaly detection, a technique used in Intrusion Detection, which uses behaviour based approaches to detect abuse and misuse of computer systems. Traditional approaches to computer security have relied on signature based approaches for the detection of intruders. Network based intrusion detection systems (IDS) such as Snort [25] cross reference patterns of network packets against a database of known intrusions. If a packet matches any of the signatures contained in the database an alert is generated, notifying the user of a potential intrusion. One problem with signature based approaches is that slightly modified intrusions or brand-new intrusions are not detected as they are not contained within the database resulting in false negatives.

Anomaly detection offers an alternative approach, by using a defined database of 'normal', either in terms of machine behaviour or user behaviour. Data at run time is compared against the normal profile and sufficient deviation causes the generation of alert. This is demonstrated through the research of the negative selection algorithm [13] which forms the majority of anomaly detection research within artificial immune systems. Unfortunately, defining what is normal is non-trivial and has a tendency to change over time, giving rise to systems with a high rate of false positives. To overcome the problems of false positives, a whole host of methods have been employed. This frequently involves adding a dynamic profiler to account for expected changes in the normal profile, or the use of more and disparate data sources. It is worthy of note that

In computer security, anomaly detection has been applied to a wide range of problems. This includes the detection of trojans, viruses, rootkits, network exploits, and distributed denial of service. As an application of anomaly detection in computer security, we examine the problem of detecting port scans. They are a key tool in initiating an attack, and are frequently used in 'insider attacks' which are performed by authorised users.

2.1. General principles of port scanning

Port Scanning is a technique of network cartography. It is used by system administrators to check specified hosts on their network for availability and to monitor services in use. However it can be subverted for more malicious purposes. Port scanning tools such as 'Network Mapper' (`nmap`) [23] can reveal information about hosts responding on a given set of network addresses. This information may be used by attackers to discover a set of target hosts which are operating services likely to be vulnerable to attack. It can also be used for an attacker to learn and understand the topology of a network in order to launch an attack such as a distributed denial of service.

A host on an IP network has one or more IP addresses. Each IP address has a range of 2^{16} TCP ports and 2^{16} UDP ports. Ports are simply a way of multiplexing many different types of communication through a single network address. This is why it is possible to download mail and surf the web at the same time. A program running on a network host may listen for requests on one or more (address, protocol, port) tuples. Many services typically listen for requests on standard port numbers (such as TCP port 80 for the HTTP service), though in reality, a service can be located on any port number. Port scanning involves probing a host to discover potential exploitable ports.

Instances of port scans differ from each other through a number of important properties. At a high level of abstraction there are two distinctions to be made. Firstly, modern networks are comprised of suites of various network protocols which offer different kinds of endpoints which can be useful to map. This work restricts discussion to IP networks in which there are IP address, TCP port and UDP port endpoints. Scans which map out these different types of endpoints use different methods. Secondly, attackers rarely wish to scan every possible endpoint reachable from their network, so a subset of endpoints are selected for mapping. The way in which the scope of the scan is restricted leads to a differing '*scan footprint*'.

Once a scanner has created a list of endpoints, a probe is performed on each endpoint in order to obtain the scan results. As mentioned, various probe techniques are available depending on what kind of endpoint is being probed and for what information. The three main types of probe are:

(1) Host probe: Determining if a given network address is assigned to a host.
(2) Port probe: Determining if a service is listening at an (address, protocol, port) tuple.
(3) Service probe: Determining what kind of service is running over an (address, protocol, port) tuple.

Host probes are typically carried out by sending ICMP echo requests to the IP address being queried. For this reason host scans are usually referred to as "ping scans" after the name of the UNIX program for sending these packets. If a host is associated with the queried IP address it may respond with an ICMP echo reply. However many systems simply do not respond to echo requests due to the potential for abuse. For this reason TCP probes may be sent to a port likely to be unfiltered (such as TCP port 80) and any response at all from that address is considered positive. If there are intervening routers between the scanner and the target host an ICMP host unreachable message may be generated for any traffic sent to an inactive address.

In the TCP/IP protocol suite, UDP and TCP port probes are possible. TCP port scans occur with much higher frequency than other types of scan. The simplest type of TCP probe connects to a port on a remote address and if the connection succeeds immediately closes the connection. A more stealthy approach, termed a "SYN scan" simply sends the first packet of the three-way handshake and uses the response packet to distinguish between open and closed ports. This usually requires super-user privileges. The only available technique for probing UDP ports is to send a packet containing random data to a UDP port on a remote host. If the port is open no response will be generated and if the port is closed an ICMP port unreachable error message is generated. Service scans are typically carried out after a port scan and lead to knowledge of the type and version of operating system and network service software running on a remote host. In fact, the exact behaviour elicited by a host as a response to any of these probe types can be used in determining the operating system type and version.