A Conceptual Space Model for Intrusion Detection

Brian J. d'Auriol
Department of Computer Science
The University of Texas at El Paso
El Paso, TX, USA 79968
dauriol@acm.org

Afolami Akinsanmi
Microsoft Corporation
fola@microsoft.com

***Abstract*— *This paper describes an approach for intrusion detection that is based on conceptual representations. The Conceptual Space model is adapted to represent information about potential intrusive activity. In order to more easily provide for decisions regarding assessment of intrusive behavior, we incorporate superellipsoids into our adapted conceptual space model. Brief comments regarding conceptual spaces and superellipsoids are made. We illustrate our approach by considering a network intrusion detection application.*

Keywords: Intrusion detection, Computer security, Conceptual space.

I. Introduction

Computer systems are valuable resources of any organization and there is a need to ensure that these resources are not compromised. The potential impacts or effects of a compromise include unavailable resources, financial loss, information destruction and information theft. Given the history of computer attacks that result in system compromise, it is reasonable to assume that intruders will persist in attacking computer resources. In particular, systems connected to the Internet face greater risk of intrusion as they are more accessible [1], [2].

Intrusion Detection Systems (IDS) detect malicious or unauthorized activities on a computer host. The IDS reports on the level of intrusion activity on computer resources by issuing alerts and notifications to system managers. Often, the IDS also includes routines that actually respond to these intrusions. Early work in intrusion detection includes Anderson's in 1980 [3] and Dennings' in 1987 [4]. The two main kinds of intrusion detection systems are anomaly based and misuse (signature) based. The former often employ statistical or other means to classify normal from abnormal behavior whereas the latter use signature files, rules or other static model-based approaches to assess behavior. Both types of systems have advantages and disadvantages. Implementations can have host based or network-based sensors and systems. A number of surveys have been published, see for example [1], [5].

This paper describes an approach for intrusion detection that is based on conceptual representations. The Conceptual Space model [6] is adapted to represent information about potential intrusive activity. In order to more easily provide for decisions regarding assessment of intrusive behavior, we incorporate superellipsoids into our adapted conceptual space model. Brief comments regarding conceptual spaces and superellipsoids are made. We illustrate our approach by considering a network intrusion detection application.

The proposed conceptual space model described in this paper forms the basis of the Conceptual Layer described in the Network Vertical Intrusion Model (NetVIM) [7]. NetVIM is a vertical four layer model that provides for the identification, detection and profiling of network based attacks. Figure 1 describes these layers.

This paper is organized as follows. Conceptual spaces are reviewed in Section II and superquadrics are reviewed in Section III. The model for intrusion detection based on conceptual spaces and superquadrics is presented in Section IV. An intrusion detection application of our model is presented in Section V. Some comments regarding general application of the model are given in Section VI. Conclusions are given in Section VII.

II. Conceptual Spaces

Conceptual Spaces, recently proposed by Gärdenfors [6], uses a geometric framework to represent information about a modeled entity at the conceptual level. A conceptual space is a collection of one or more quality domains, along with information on how these domains are related. A quality domain is a set of integral quality dimensions that are separable from all other dimensions. A quality dimension is used to represent an attribute or property of the modeled entity.

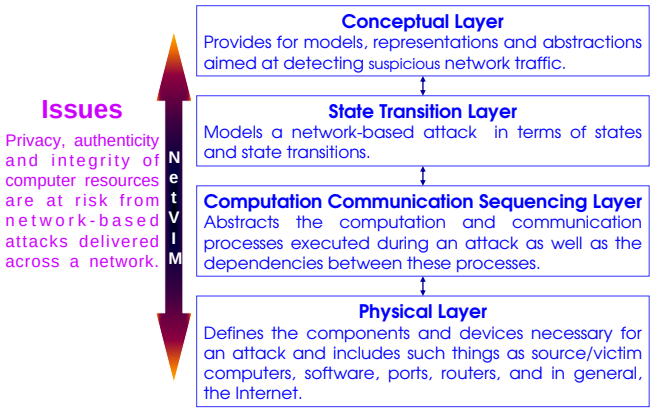

Fig. 1. The Network Vertical Intrusion Model (NetVIM)

Each quality dimension is endowed with a geometric structure and is also a metric space.

Quality dimensions allow similarity between objects to be modeled implicitly. Similarity between objects in a conceptual space is defined as an exponentially decaying function of the distance between their representing points in the space. Let d_{ij} be the distance between two objects i and j. Then the similarity between the objects, s_{ij}, can be expressed by the following relation $s_{ij} = e^{-c \cdot d_{ij}}$ where c is a constant. Quality dimensions may be integral or separable. Two dimensions are integral if an object cannot be assigned a value in one dimension without being given a value in the other dimension. Dimensions that are not integral are separable.

Information is represented in terms of concepts and properties. A natural concept is represented as a set of regions in a number of domains, together with an assignment of salience weights to the domains. A natural concept also includes information on how the regions in different domains are correlated. Operations on concepts may be defined, for example, concept combination. A natural property is a convex region of a domain in a conceptual space. A region C of a conceptual space is described as convex if for all points x, y in C, all points between x and y are also in C.

In this paper, a single domain representing an intrusion is constructed. The quality dimensions represent the security attributes that are relevant for detecting intrusions.

III. Superquadrics

A superquadric is a three-dimensional geometric object constructed from parametric equations. Superquadrics were first proposed by A. H. Barr in 1981 [8]. Superellipsoids, supertoroids, and superhyperboloids of one and two pieces are special forms of superquadrics. These 3D surfaces can be obtained by the spherical product of two 2D curves. The parametric equation for a superellipsoid is given in Equation 1.

$$\mathbf{r}(\eta, \omega) = \begin{bmatrix} a_1 \cos^{\varepsilon_1} \eta \cos^{\varepsilon_2} \omega \\ a_2 \cos^{\varepsilon_1} \eta \sin^{\varepsilon_2} \omega \\ a_3 \sin^{\varepsilon_1} \eta \end{bmatrix}, \quad \begin{matrix} -\pi/2 \leq \eta \leq \pi/2 \\ -\pi \leq \omega < \pi \end{matrix} \tag{1}$$

The parameters a_1, a_2 and a_3 are the lengths of the superquadric axes; ε_1 and ε_2 are rational numbers that determine the shape of the superellipsoid and are known as form factors. The object assumes a squared shape if the form factors are less than 1, it is more rounded for values that are close to 1, and assumes a cuspidate shape for greater values. The implicit superquadric equation, analogous to Equation 1 is given in Equation 2.

$$\left(\left(\frac{x}{a_1} \right)^{\frac{2}{\varepsilon_2}} + \left(\frac{y}{a_2} \right)^{\frac{2}{\varepsilon_2}} \right)^{\frac{\varepsilon_2}{\varepsilon_1}} + \left(\frac{z}{a_3} \right)^{\frac{2}{\varepsilon_1}} = 1 \tag{2}$$

Superquadrics are used widely in computer vision because it provides a compact representation for objects with rounded edges. Superquadrics can be easily rendered, shaded, or the shape altered by modifying the parameters of its parametric equation [9]. Two properties of interest in this paper are the description of superquadrics in general position, and computing the distance between a point and a superellipsoid.

Equation 1 describes a superquadric using 5 parameters that assume a superquadric-centered coordinate system. In order to describe superquadrics in general position i.e. a global coordinate system, a transformation is applied to the Equation 1. A detailed description of this process is given in [9]. For a given point, the process first rotates that point and then translates it to the new coordinate system. To achieve this, 6 additional parameters are included in the equation of a superquadric in general position. These include 3 center-coordinates (p_x, p_y, p_z) for translating the points and 3 Euler angles (ϕ, θ, ψ) for rotating the point.

The formulas presented in here compute *radial Euclidean distance* between a point x_0, y_0, z_0 and a superellipsoid. The radial Euclidean distance is defined as the distance between a point and a superellipsoid along a line through the point and the center of the superellipsoid. Let $F(x, y, z)$, given by Equation 2 define a superellipsoid. The idea is to compute a scalar β such that $F(\beta x_0, \beta y_0, \beta z_0)$ defines a new superellipsoid

whereby the given point x_0, y_0, z_0 lies on its surface i.e.

$$F(\beta x_0, \beta y_0, \beta z_0) = \left[\left(\frac{\beta x}{a_1} \right)^{\frac{2}{\varepsilon_2}} + \left(\frac{\beta y}{a_2} \right)^{\frac{2}{\varepsilon_2}} \right]^{\frac{\varepsilon_2}{\varepsilon_1}} + \left(\frac{\beta z}{a_3} \right)^{\frac{2}{\varepsilon_1}} = 1 \quad (3)$$

From this equation, it follows that given the point x_0, y_0, z_0, its position relative to the superellipsoid can be determined by computing $F(x_0, y_0, z_0)$, which is also called the *inside-outside* function. The following properties hold

(i) $F(x_0, y_0, z_0) = 1 \iff$ the point belongs to the surface of the superellipsoid

(ii) $F(x_0, y_0, z_0) > 1 \iff$ the point is outside the superellipsoid

(iii) $F(x_0, y_0, z_0) < 1 \iff$ the point is inside the superellipsoid

IV. Model for Intrusion Detection

The conceptual space model allows representation of intrusive behaviors while superquadrics provide for the classification of a given observation as intrusive or not. The inclusion of a superellipsoid into a conceptual space conveniently allows classification via Equation 3 of the security properties represented by the quality dimensions; hence, the classification establishes inclusion or exclusion of the event into the concept of an intrusion. Prior work on the inclusion of superquadrics into conceptual spaces appears in [10] where the authors consider a model for describing actions of robots.

The concepts to be represented are the security risks of a monitored system (alternatively, non-security risks). These concepts are represented by a set of one or more bounded convex regions in a conceptual space. Boundedness and convexity are achieved through the use of superellipsoids. Domains represent set of possible states for a system that is monitored. Domains consist of a set of quality dimensions that are highly correlated. Quality dimensions represent the security attributes that are relevant for detecting intrusions. Since attribute values are obtained from sensored data, an appropriate scaling for each dimension is determined. Dimensions are identified with the spatial dimensions x, y and z of the superellipsoid. The form and scaling factors of the superellipsoid reflect the shape of the object, hence, reflect the 'shape' of the domain.

Each point in the conceptual space interior to or on the superellipsoid represents a three element vector corresponding to a potential observation of three attributes which together signify an intrusion (alternatively, non-intrusion). Given an instance of the three attributes generated from a sensor data, the event causing these instances can be characterized as appropriate behavior or an intrusion by determining if the corresponding point in the conceptual space is within or on the superellipsoid (intrusion) or external to (not an intrusion) the superellipsoid (alternatively, non-intrusion and intrusion, respectively).

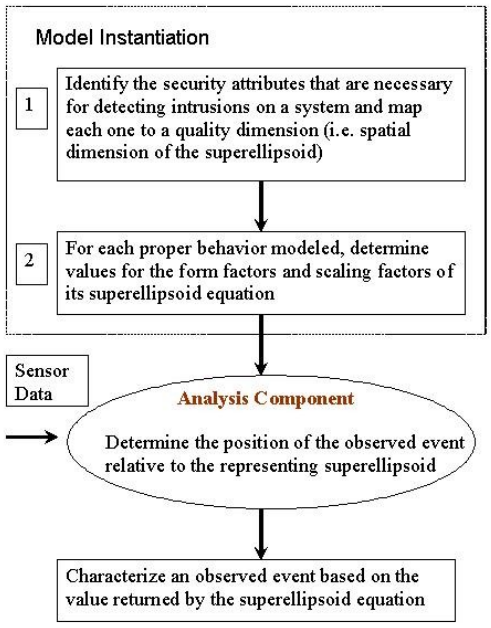

Fig. 2. Detecting intrusions in the Conceptual space

Figure 2 describes the application of the intrusion detection model. First, the conceptual model needs to be instantiated for a specific intrusion activity. This requires: 1) identifying the security attributes and quality dimension scalings and 2) determining the form and scaling factors. Second, during run-time operations, sensor data is obtained, scaled according to the quality dimension scalings and characterized as intrusion or not depending on the value returned by the superellipsoid equation.

V. Application

Characterization of network attacks is one area of much active research. In [11], the authors describe related work and analysis techniques. Port scans are characterized in [12][1]. Categorical measures for network

[1]Personal communication with the authors confirms this citation as unpublished older work; we have been made aware of work on an updated manuscript.

traffic are presented in [13]. The prototype IDS system considered in this section entails network-based intrusion detection.

An intrusion whereby an attacker attempts to obtain valid login credentials (i.e. username and password) to a Microsoft® Windows 2000 host is simulated as an illustration of the proposed approach. This simulation consists of two types of attacks, a probe attack and a dictionary attack.

A. Simulation Environment

The simulation platform consists of a victim host that runs the Microsoft® Windows 2000 operating system. This computer hosts a Microsoft .NET® Web Service. A web service is a software component that is accessible to other applications over a network through standard web protocols and exchanges data using XML. The victim computer is connected to a network monitored by the IDS. The IDS obtains copies of network data by employing a "promiscuous" mode access to the network whereby it reads data communicated via the network regardless of its destination (ordinarily, systems would only read network data that is addressed to them). Network sniffing is performed by running the *WinDump* program which returns the TCP headers of packets transmitted on a network interface that match a specified selection criteria. The general format of the TCP protocol line returned by *WinDump* is

src $>$ **dst: flags data-seqno ack window urg options**

where **src** and **dst** specifies the address and port for the source and destination of a connection; **flags** are some combination of S (SYN), F (FIN), P (PUSH) or R (RST) or a single '.' (no flags); **data-seqno** describes the portion of sequence space covered by the data in this packet; and **ack** is sequence number of the next data expected the other direction on this connection; **window** is the number of bytes of receive buffer space; **urg** indicates there is 'urgent' data in the packet; and **options** are TCP options specified in the packet [14].

B. Attack Description

The probe attack is the first stage in the simulation. The attacker runs a program that obtains NetBIOS information from the victim computer. Simulations in this paper implemented this probe attack by running the program *nbtdump.exe* and specifying the name of the victim host as a parameter. If the program is successful, it returns share information consisting of the share name, its type, and a descriptive comment about the share. It also returns account information consisting of account names as well as limited password information about these accounts such as the number of days since the password was changed and number of time the account has been used to logon to the system. This information is saved in an html file named *victim*.html where *victim* is the name of the victim host specified as a parameter to the program.

The second stage of the attack is a dictionary attack whereby the attacker guesses a password, and using account names obtained earlier, tries to logon to the victim host. The guessed password could be obtained using a brute force method, or by doing a dictionary lookup. The simulations in this paper read the password from a password dictionary as this makes the simulations more manageable.

A Microsoft® Windows 2000 host can be configured to disable an account once a maximum number of failed logins is exceeded. In the research, it has been determined that to circumvent this restriction, the attacker attempts to connect to the victim host using a service that supports authentication via usernames and passwords. In the simulation, the *net use* command supplied by the operating system is employed. The connection is established successfully only if correct credentials are specified. Failed login attempts do not count towards disabling an account and hence, the program is used for the dictionary attack.

C. Attack Detection

A dictionary attack is characterized by a large number of failed login attempts. In the simulation described, the attacker is identified by the **src** field in the *WinDump* data. This is a necessary signature since an attacker that tries to establish a connection using a fake address would have no way of knowing the status of the requested connection as the victim host sends its packets to the spoofed address.

An analysis of the WinDump data for requested connections reveal that approximately 600 bytes of data are communicated to the victim host to request a connection. The similarity of the observed event to this attribute is mapped to the x quality dimension. Connection requests tend to have a lower rate of data packets per connection and this attribute is mapped to the y quality dimension. The analysis also reveals a difference in the patterns between successful connections and other connections. When a successful connection is established, the initiator always sends an acknowledgement packet with no flags

```
20:31:32.562947 XXXX.cs.zzz.edu.1277 > yyyy.cs.zzz.edu.445: F
        600:600(0) ack 368 win 63873 (DF)
```

Fig. 3. Failed connection.

```
20:31:27.764700 XXXX.cs.zzz.edu.1149 > yyyy.cs.zzz.edu.445: .
        ack 510 win 63731 (DF)
```

Fig. 4. Successful connection.

or data back to the victim host. When a connection fails, a packet with the *FIN* flag set is sent to the victim immediately following communication of the logon credentials. This attribute is mapped to the z quality dimension. For example, Figure 3 shows a captured TCP header returned by *WinDump* that indicates a failed connection which is closed abruptly while Figure 4 shows a captured TCP header that indicates the completion of a successful connection and data transmission

Another feature of this attack is that the connection is always requested on port 445 of the victim computer. In Windows® 2000, Microsoft has created a new transport for SMB (Server Message Block) over TCP and UDP on port 445. SMB is a protocol for sharing files, printers, serial ports, and communications abstractions such as named pipes and mail slots between computers [15]. Hence, the requested connection is via SMB messages communicated between port 445 of the victim computer and an attacker.

D. Implementation

A program to simulate the dictionary attack described earlier is developed. The program is object oriented and includes a graphical user-interface for easy configuration and attack setup. This program takes as input the .html file returned from a probe attack. Connections to the victim host are requested using usernames and share names extracted from the file. An IDS that implements the proposed approach is also developed. Source codes are written in Microsoft C#.NET®. Code fragments are available in [16].

The simulation consists of five sets of experiments. TCP headers from the network traffic generated by these simulations are recorded and passed as input to the IDS prototype. The simulations are as follows:

Simulation 1 Run *nbtdump.exe* with the computer name or address of the victim computer specified as a command line parameter to the program. This probe attack returns NetBIOS information about the victim computer.

Simulation 2 Execute an application that resides on the attacker's computer. The application consumes the web service hosted on the victim computer. This simulation does not contain any intrusion activity.

Simulation 3 Run the *Password Guessing Application* and try to obtain valid credentials on the victim host. If a successful connection is established, issue a notification to the attacker and close the connection.

Simulation 4 Run Simulations 2 and 3 simultaneously so that the attack is masked by network traffic generated as a result of calls to the web service hosted on the victim computer.

Simulation 5 Run the *Password Guessing Application* and try to obtain valid credentials on the victim host using a smaller dictionary that contains 1000 words.

E. Parameter Selection

Spatial variables x, y, z in the superellipsoid equation are mapped to quality dimensions of the conceptual space. The variable x denotes the similarity between the current connection and a network connection request based on the number of bytes transmitted. The variable y represents the similarity to a network connection request based on the ratio of data packets in the connection. The variable z represents the failure rate for network connection requests.

The values for form factors and scaling factors are selected through a combination of expert knowledge, simulation and experimental fine tuning that is described below. These values are chosen in such a way that the resulting superellipsoid represents only proper behavior by points that lie within the geometric object. The final values used in the simulation are as follows:

$$\begin{aligned} &\varepsilon_1 = 2.5, \qquad \varepsilon_2 = 1.25, \\ &a_1 = 2.5, \qquad a_2 = 3.0, \qquad a_3 = 2.25 \end{aligned}$$

These parameters describe a superellipsoid whose shape is shown in Figure 5. The parameters ε_1 and ε_2 determine the shape of the superellipsoid. For values greater than 1, the object takes on a cuspidate shape. This is noticed along the z dimension. Values close to 1 generate a rounded shape, noticed here along the xy-plane. Intuitively, since points inside of the superellipsoid characterize valid behavior, the cuspidate shape of the superellipsoid along the z dimension indicate that an observation is more likely to be characterized as an intrusion if it has a high rate of login failures. Values for each of the scaling factors indicate the threshold beyond

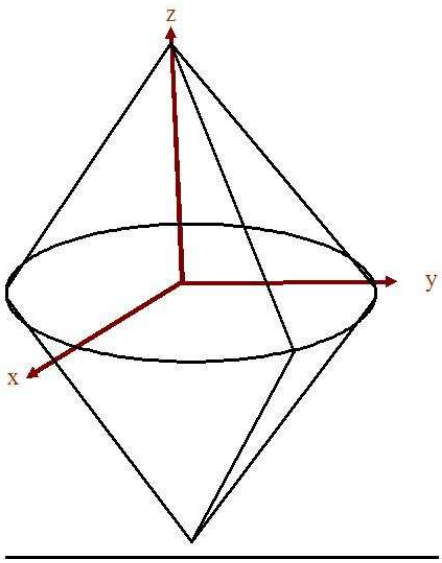

Fig. 5. Superellipsoid representation for the attack simulation

which an observation is characterized as an intrusion in these simulations. Note that other values could be reasonably chosen. However, for poorly chosen values, inaccurate characterizations may occur.

F. Simulation Results

Results from the simulations are shown in Table I. The final output of the IDS is the value returned by the superellipsoid equation. Four of the simulations are correctly identified as intrusions and have values greater than 1. For these simulations, the represented points in the conceptual space lie outside the superellipsoid. Even when the attack was masked by calls to the web service, the attack was still detected even though its representational point is slightly closer to the superellipsoid. The probe attacks in Simulation 1 were also detected. Valid program behavior, such as accessing a web service in Simulation 2 on the victim host, is correctly identified as its associated value is less than 1.

TABLE I

RESULTS RETURNED BY THE IDS FOR SIMULATIONS CONTAINING DICTIONARY ATTACKS, A PROBE ATTACK AND ACCESSING A WEB SERVICE ON THE VICTIM HOST

Simulation	x	y	z	$\mathbf{F}(x, y, z)$
1	0.45	0.81	0.86	1.24
2	0.07	0.13	0.12	0.51
3	0.62	0.94	0.99	1.37
4	0.62	0.87	0.96	1.34
5	0.66	0.89	0.98	1.36

From the results, the system shows good performance in detecting attacks. The rate of false positives and false negatives is 0%. For each simulation, the value returned by the superellipsoid equation should be interpreted in terms of the semantics of the proposed conceptual representation. The value represents the amount of intrusion activity observed in the simulation. Compared to simulation 2 for example, the IDS returns a lower value when simulations 2 and 3 are run simultaneously. Intuitively, this behavior is expected since the proper behavior in simulation 2 reduces the ratio of intrusion activity relative to the total observed activity.

VI. GENERAL APPLICATION

Different attack types can be detected by instantiating the model as describe in Figure 2. If a proper behavior is associated with less that three attributes, it can be represented in this model by setting the values along dimensions that are not considered to the identity. For example, a proper behavior associated with two attributes is represented by setting $\varepsilon_1 = 1$, $a_3 = 1$ and $z = 0$. For proper behaviors that are associated with more than three attributes, the model is still applicable. The restriction in this model is on the maximum number of attributes that can be correlated within a domain and not on the maximum number of attributes that may be associated with a proper behavior. In this case, the conceptual space will consist of more that one domain, each of which is represented by a superellipsoid. The output of the IDS is obtained as an aggregation of results returned by each of the superellipsoid representations.

VII. CONCLUSIONS

A model for intrusion detection that is based on conceptual representations has been described in this paper. We first have described the conceptual space model which forms the basis of our work. We have enhanced the conceptual space model by incorporating superellipsoids. And we have proposed an algorithm that can be deployed in an IDS based on our model. A prototype IDS based on our model is described. Lastly, we have deployed the prototype IDS in an experimental study. The experiments together with the prototype system succeeded in showing that our model is practical and useful. However, further work remains, including, additional evaluation of the prototype IDS by using standardized tests, for example, that which is available at [17].

REFERENCES

[1] J. S. Sherif and T. G. Dearmond, "Intrusion detection: Systems and models," in *proc. of the Eleventh IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE'02)*, 2002, pp. 1–19.

[2] "Csi/fbi computer crime and security survey," 2003, http://www.gocsi.com/.

[3] J. P. Anderson, "Computer security threat monitoring and surveillance," James P. Anderson Company, Fort Washington, PA, Tech. Rep. 79F296400, February 1980.

[4] D. E. Denning, "An intrusion-detection modelj," *IEEE Transactions on Software Engineering*, vol. SE-13, no. 2, pp. 222–232, 1987.

[5] J. Allen, A. Christie, W. Fithen, J. McHugh, J. Pickel, and E. Stoner, "State of the practice of intrusion detection technologies," Pittsburgh, PA 15213-3890, Tech. Rep. CMU/SEI-99-TR-028, Jan. 2000.

[6] P. Gärdenfors, *Conceptual Spaces, The Geometry of Thought*. Cambridge, MA: MIT Press, 2000.

[7] B. J. d'Auriol, "Network vertical intrusion model (NetVIM)," in *Proc. of The 2005 International Conference on Security and Management (SAM'05)*, Monte Carlo Resort, Las Vegas, Nevada, USA, June 2005, in press.

[8] A. H. Barr, "Superquadrics and angle-preserving transformations," *IEEE Computer Graphics and Applications*, vol. 1, no. 1, pp. 11–23, January 1981.

[9] A. Jaklič, A. Leonardis, and F. Solina, *Segmentation and recovery of superquadrics*. Boston, MA, USA: Kluwer Academic Publishers, 2000.

[10] A. Chella, S. Gaglio, and R. Pirrone, "Conceptual representations of actions for autonomous robots," *Robotics and Autonomous Systems*, vol. 34, pp. 251–263, 2001.

[11] P. Barford and D. Plonka, "Characteristics of network traffic flow anomalies," in *ACM SIGCOMM Internet Measurement Workshop 2001*, San Francisco, Nov. 2001.

[12] C. B. Lee, C. Roedel, and E. Silenok, "Detection and characterization of port scan attacks (manuscript, believed 2002)," http://www.cse.ucsd.edu/~esilenok/portscans.html, University of California, San Diego.

[13] P. Porras and A. Valdes, "Live traffic analysis of TCP/IP gateways," in *Proceedings of the 1998 ISOC Symposium on Network and Distributed System Security*, March 1998, . [Online]. Available: http://www.sdl.sri.com/projects/emerald/live-traffic.html

[14] *WinDump Manual*, March 2002. [Online]. Available: http://windump.polito.it/docs/manual.htm

[15] R. Sharpe, "Just what is SMB?" Oct. 2002, maintained by samba: http://www.samba.org. [Online]. Available: http://samba.anu.edu.au/cifs/docs/what-is-smb.html

[16] A. Akinsanmi, "A conceptual space model for intrusion detection," Master's thesis, Department of Computer Science, The University of Texas at El Paso, December 2002.

[17] M. Zissman, "DARPA intrusion detection evaluation," 2001, lincoln Laboratory. [Online]. Available: http://www.ll.mit.edu/IST/ideval/index.html