A Methodology for Using Intelligent Agents to provide Automated Intrusion Response

Curtis A. Carver, Jr., John M.D. Hill, John R. Surdu *Member, IEEE,* and
Udo W. Pooch, *Senior Member, IEEE*

***Abstract*--This paper proposes a new methodology for adaptive, automated intrusion response (IR) using software agents. The majority of intrusion response systems (IRSs) react to attacks by generating reports or alarms. This introduces a window of vulnerability between when an intrusion is detected and when action is taken to defend against the attack. Research by Cohen indicates that the success of an attack is dependent on the time gap between detection and response. If skilled attackers are given ten hours after they are detected and before a response, they will be successful 80% of the time. At thirty hours, the attacker almost never fails [1]. The proposed methodology addresses this window of vulnerability by providing an automated response to incidents using a heterogeneous collection of software agents. These agents collaborate to protect the computer system against attack and adapt their response tactics until the system administrator can take an active role in the defense of the system.**

***Index Terms*– Intrusion Response, computer security, intelligent agents**

I. INTRODUCTION

THE number of information warfare attacks is increasing and becoming increasingly sophisticated. Annual reports from the Computer Emergency Response Team (CERT) indicate a significant increase in the number of computer security incidents each year. Figure 1 depicts the rise the computer security incidents with six incidents reported in the 1988 and 8,268 in 1999 [2]. Not only are these attacks becoming more numerous, they are also becoming more sophisticated. The 1998 CERT Annual Report reports the growing use of "widespread attacks using scripted tools to control a collection of information-gathering and exploitation tools" [3]. The 1999 CERT Distributed Denial of Service Workshop likewise reports the growing use of automated scripts that launch and control tens of thousands of attacks against one or more targets. Each attacking computer has limited information on who is initiating the attack and from where [4]. The threat of a sophisticated computer attacks is growing. Unfortunately, intrusion detection and response systems have not kept up with the increasing threat.

Current intrusion detection systems (IDSs) have limited response mechanisms that are inadequate given the current threat. While IDS research has focused on better techniques for intrusion detection, intrusion response remains principally a manual process. The IDS notifies the system administrator that an intrusion has occurred or is occurring and the system administrator must respond to the intrusion. Regardless of the notification mechanism employed, there is a delay between detection of a possible intrusion and response to that intrusion.

This delay in notification and response, ranging from minutes to months, provides a window of opportunity for attackers to exploit. Cohen explored the effect of reaction time on the success rate of attacks using simulations [1]. The results indicate that if skilled attackers are given ten hours after they are detected before a response, they will be successful 80% of the time. If they are given twenty hours, they will succeed 95% of the time. At thirty hours, the attacker almost never fails. The simulation results were also correlated against the skill of the defending system administrator. The results indicate that if a skilled attacker is given more than thirty hours, the skill of the system administrator becomes irrelevant - the attacker will succeed. On the other hand, if the response is instantaneous, the probability of a successful attack against a skilled system administrator is almost zero. Response is a fundamental factor in whether or not an attack is successful. For the response to be successful against skilled attacker, the response system must adapt its tactics so that the response system does not always respond with a static defense. Attackers would simply adapt their approach so as to mediate the defense. An adaptive, automated intrusion response system provides the best possible defense and shortens or closes this window of opportunity until the system administrator can take an active role in defending

All authors are with the Department of Computer Science, Texas A&M University, College Station, Texas 77843 (email: {carverc, hillj, surdu pooch}@cs.tamu.edu).

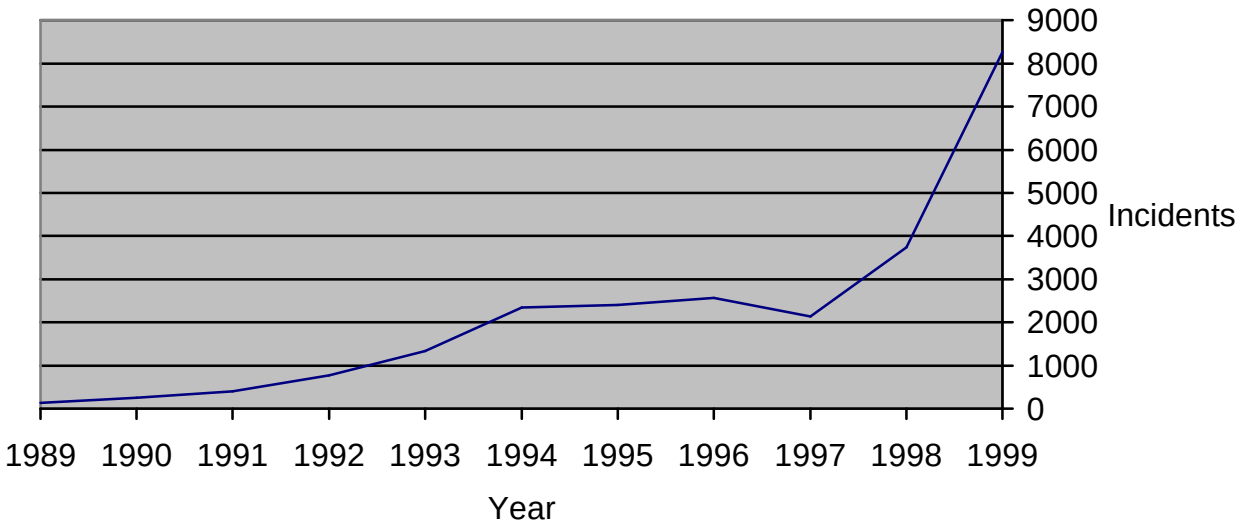

Figure 1: CERT Reported Incidents per Year

against the attack. Unfortunately, no such response system exists.

II. Previous Work

In the past seventeen years, there have been a number of intrusion detection and intrusion response tools developed (See Table 1). The response systems can be categorized as notification systems, manual response systems, or automatic response systems. The majority of intrusion detection and response systems are notification systems only - systems that generate reports and alarms only. Some systems provide the additional capability for the system administrator to initiate a manual response from a limited preprogrammed set of responses. While this capability is more useful than notification only, there is still a time gap between when the intrusion is detected and when a response is initiated. Automatic response systems immediately respond to an intrusion through pre-programmed responses. With two exceptions, all of these automatic response systems use a simple decision table where a particular response is associated with a particular attack. If the attack occurs, the preprogrammed response executes. This preprogrammed response was predominantly the execution of a single command or action instead of the invocation of a series of actions to limit the effectiveness of the attacker. The two exceptions are the Cooperating Security Managers (CSM) and Event Monitoring Enabling Responses to Anomalous Live Disturbances (EMERALD).

Intrusion Response Classification	# of Systems
Notification	31
Manual Response	8
Automatic Response	17
Total	56

Table 1: Classification of Intrusion Response Systems

Cooperating Security Managers (CSM) is a distributed and host-based intrusion detection and response system. CSM proactively detects intrusions without using a central director. CSM provides responses through three different components: the Command Auditor, the Damage Control Processor (DCP), and the Damage Assessment Processor (DAP). The Command Auditor examines the user's command stream and automatically discards commands that it identifies as an attack. The DCP reactively responds to intrusive behavior using the Fisch DC&A taxonomy to classify the attack as well as the suspicion level assigned to the user by the intrusion detection system. As the suspicion level changes, the DCP employs eight different response sets, each of which consists of one or more of fourteen different response actions. DCP continues to respond to intruder actions until the intruder leaves the system when their suspicion level is reset to zero. After the intruder leaves, the DAP attempts to restore the system to its pre-attack state [5, 6]. CSM does not maintain any state information concerning an ongoing attack other than a suspicion level. This lack of state information limits the effectiveness of the system. By resetting the suspicion level

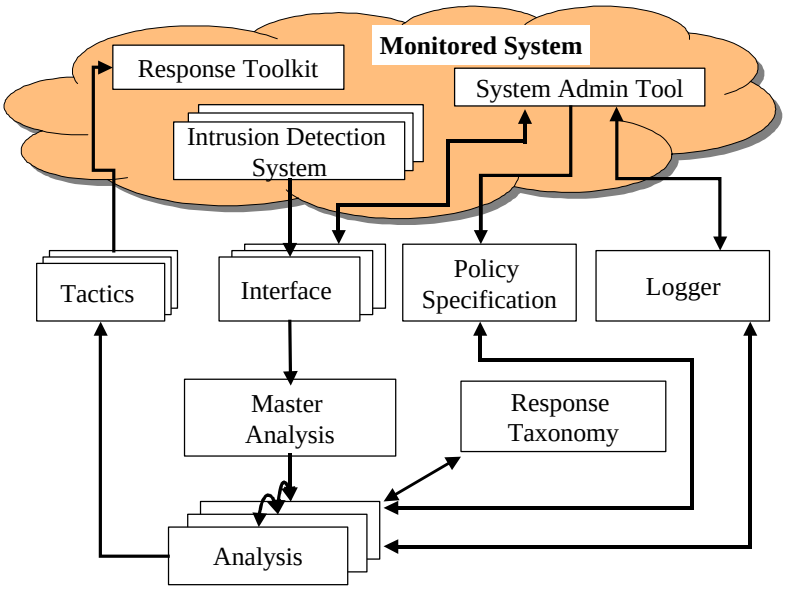

Figure 2: Proposed Methodology

to zero after the intruder leaves the system, the attacker has the opportunity to begin their attacks anew. CSM also does not adapt its responses based on success of previous responses or the accuracy of the intrusion detection system nor does it have any mechanism for generating a course of action other than a preprogrammed action associated with a particular suspicion level. This research addresses these open research issues.

Event Monitoring Enabling Responses to Anomalous Live Disturbances (EMERALD) is a distributed misuse and anomaly intrusion detection system. It is intended for large-scale heterogeneous computing environments. The EMERALD architecture consists of hierarchical collections of monitors. There are four principal components in the each monitor: Profiler Engine, Signature Engine, Resource Object, and Resolver. The Profiler Engine is a statistical anomaly detection component of the system while the Signature Engine is the signature-based inference component. The Resource Object is a pluggable configurable library with all of the data for the other three components. Finally, the Resolver is the coordinator of analysis and response policy enforcer. Every monitor has an intrusion response capability. The resolver is an expert system that receives reports from the analysis components and invokes various response handlers. The possible responses are defined in the resource object with two associated metrics that delimit their usage: a threshold metric and a severity metric. The threshold metric defines the degree of intrusive evidence necessary to use the response. The severity metric defines how harsh a particular response is [7, 8]. EMERALD response mechanism is limited in its use of two metrics to determine an appropriate response. This research extends the EMERALD approach to provide a more formal and robust methodology for intrusion response that incorporates a more complete response taxonomy and an adaptive response mechanism that utilizes a number of additional criteria in formulating an appropriate response.

III. DISCUSSION

A. *Overview*

The proposed methodology is summarized in Figure 2. Multiple IDSs monitor a computer system and generate intrusion alarms. *Interface agents* maintain a model of each IDS based on number of false positives/negatives previously generated. It uses this model to generate an attack confidence metric and passes this metric along with the intrusion report to the *Master Analysis agent*. The *Master Analysis agent* classifies whether the incident is a continuation of an existing incident or is a new attack. If it is a new attack, the *Master Analysis agent* creates a new *Analysis agent* to develop a response plan to the new attack. If the incident is a continuation of an existing attack, the *Master Analysis agent* passes the attack confidence metric and intrusion report to the existing *Analysis agent* handing the attack. The *Analysis agent* analyzes an incident until it is resolved and generates an abstract course of action to resolve the incident. To generate this course of action, the *Analysis agent* invokes the *Response Taxonomy agent* to

classify the attack and *Policy Specification agent* to determine a response goal and limit the response based on legal, ethical, institutional, or resource constraints. The *Analysis agent* passes the selected course of action to the *Tactics agent*. The *Tactics agent* decomposes the abstract course of action into very specific actions and then invokes the appropriate components of the *Response Toolkit.* Both the *Analysis and Tactics agents* employ adaptive decision-making based on the success of previous responses. The *Logger* records *Analysis* and *Tactics agents'* decisions for system administrator review. Each of these system components are discussed in greater detail below.

B. Interface Agents

The Interface agent performs two functions: it translates IDS specific messages into generic message format and it maintain a confidence metric on the reporting IDS. There are two techniques for message formats: a general communications language such as the Knowledge Query and Manipulation Language (KQML) or Common Intrusion Detection Format (CIDF); or, the use of specialized language [9]. Because the proposed architecture must interact with a variety of different IDSs, there is no assumption of a common communications language. The Interface agent provides this translation service so that all messages internal to the response system are in a common format. Additionally, due to the requirement for rapid intrusion response, a specialized language is used internal to the response system instead of a generalized language. In an intrusion response system, the efficiency and speed a specialized language provides is more important than the flexibility and interoperability that a generalized language provides.

The Interface agent also maintains a confidence metric on the reporting IDS. IDSs are not perfect and will generate false positive and false negative alarms. The response must be tailored by the degree to which the response system believes that the reported incident is a real attack and not a false alarm. The confidence metric is the ratio of false positive reports to actual reports. The number of false positives is generated through a feedback loop between the interface agent and the system admin tool. After each incident, the system administrator can indicate whether the incident was a real attack or a false alarm. This results in an update to the confidence metric for the reporting IDS and over time, response adaptation. Responses to incidents from IDSs that generate a high number of false positives will be less severe than reports from IDSs that seldom generate false alarms.

C. Master Analysis Agent

The Master Analysis agent examines the incident report generated by the interface component and determines whether the incident is a new attack or a continuation of an existing attack. If the incident is a new attack, the Master Analysis agent creates a new analysis component and passes to it the incident report and associated confidence metric. If the incident is the continuation of a previously detected attack, the Master Analysis agent simply forwards the incident report and associated confidence metric to the appropriate analysis component.

The classification of the incidents or existing requires reasoning under uncertainty. Attackers can use a variety of techniques to hide their identity and these techniques are equally effective at confounding incident classification. Some incidents such as multiple attacks from the same Internet Protocol (IP) address within a short interval of time using the exact same program provide a clear indication of the continuation of an existing attack. Attacks such as a distributed port scan from multiple IP addresses over several days or weeks would be much more difficult to detect. In determining if an attack is a continuation of the same attack or a new attack, the Master Analysis agent uses two-level fuzzy rule set which considers the time between the incident reports, IP addresses, the user name, and the program name to determine whether an incident is a continuation of an existing attack or a new attack. Fuzzy rule bases have the advantages that it is relatively easy to capture the knowledge of domain experts and later verify how the Master Analysis agent reached classification decisions [10].

It is not the intent of this research to impose a particular inference mechanism within the Master Analysis agent but instead to advocate that there must be a classification of incidents. Repeated attacks from the same attacker require different responses than attacks from new attackers. Over the course of an attack, the response system can adapt the tactics and responses utilized to thwart an attacker. Without a classification of incidents, response adaptation is limited.

D. Analysis Agent

The Analysis agent provides long-term analysis of an incident and determines a plan, at an abstract level, to respond to an intrusion. This plan consists of a *response goal*, one or more *plan steps*, and associated *tactics* for accomplishing the plan steps. The *response goal* is specified by the system administrator and provides a general response approach. Examples of response goals include: catch the attack, analyze the attack, mask the attack from users, sustain service, maximize data integrity, maximize data

confidentiality, or minimize cost. *Plan steps* are techniques for accomplishing a response goal. Examples of plan sets include: gather evidence, preserve evidence, communicate with the attacker, slow the attack, identify compromised files, notify the system administrator, or counterattack the attacking system. *Tactics* are methods for carry out a plan step. For example, given a plan step of gather evidence, there are a variety of tactics for accomplishing this plan step such as enabling additional logging, enabling remote logging, enabling logging to an unchangeable media, enabling process accounting, tracing the connection, communicating with the attacker, or enabling additional IDSs. The tactics can be further decomposed into a number of *implementations* that are environment dependent. As an example, consider a subnet consisting of the machines Limbo, Saint Peter, and Heaven. If Saint Peter is attacked, the tactic of remote logging could be implemented by logging to computer system Limbo or Heaven or both. The analysis agent determines what plan steps and tactics are appropriate while the tactics agent determines suitable implementations.

The analysis agent develops a plan using several inputs:

- Response Goal: The Analysis agent receives a response goal from the Policy Specification agent as specified by the system administrator. The response goal influences the selection of plan sets and tactics by providing a ranking of how supportive each plan step or tactics is to the goal.

- Confidence metric: The Analysis agent receives the confidence metric from the Master Analysis agent and uses the metric so that the severity of the tactic employed is tempered by the degree of belief that an intrusion is actually occurring.

- Incident report: The Analysis agent receives the incident report from the Master Analysis agent and forwards it to the Response Taxonomy agent. The Response Taxonomy agent uses this information to classify the type of attack and type of attacker.

- Incident history: The Analysis agent maintains a history of the incident and forwards this history to the Response Taxonomy agent for proper classification. The type of attacker dimension, for example, depends on history of attacks attributed the attacker. The Analysis agent maintains this information and provides to the Response Taxonomy agent as needed.

- Success metric: The analysis agent maintains a success metric for each plan step and tactic. This success metric is the ratio of successful responses to an intrusion to the total number of responses using a particular plan step or tactic. This metric is updated after each attack by the system administrator and allows the system to dynamically adjust what plan steps and tactics are selected to respond to an intrusion. Those plans that are more successful are weighted so that they will be used more often than plan that the system administrator determines were not successful.

- Tactics adaptation: The Analysis agent coordinates with the Tactics agent to determine if the Tactics agent can implement the same tactic as previously implemented but use a different implementation.

- Policy specification: The Analysis agent coordinates with the Policy Specification agent to ensure that the plan steps and tactics being pursued is in compliance with the policy restrictions of the computing environment. These restrictions include legal, ethical, institutional, and resource-based constraints.

Given these inputs, the Analysis agent determines if a new plan is required or if it has an existing response plan. If a new plan is required, the Analysis agent invokes the Response Taxonomy agent to generate a *plan shell*, a listing of all viable plan steps and tactics based on the Response Taxonomy agent's classification of the attack. The Analysis agent then invokes the Policy Specification agent to filter the plan shell and eliminate any plan steps or tactics that are not appropriate due to legal, ethical, institutional, and resource constraints. It also uses the confidence metric to filter plan steps and tactics that are too severe given the response system's confidence in the reporting intrusion detection system. The plan steps and tactics that remain viable after this filtering process form the *plan set*. The response plan is built by selecting one or more tactics for each viable plan step using the response goal weighting of tactics and each tactic's success metric. The resulting tactics then form the *response plan* which is passed to the Tactics agent for implementation.

If there is an existing response plan, the Analysis agent reevaluates the plan to determine if it has been successful. If it has successful, then the agent can continue to execute the same plan. If the Analysis agent determines the plan has not been successful, then it can adapt by changing the plan steps employed, changing tactics, or requesting the Tactics agent to change its implementation of a tactic. If all plan steps, tactics, and implementations have been exhausted and there are indications that the system has been or is about to be compromised, the Analysis agent will shut down the host until the system administrator can take an active role in the defense of the system.

E. Response Taxonomy Agent

The Response Taxonomy agent receives input from the Analysis agent and determines the subset of plan steps and tactics that are appropriate for responding to the intrusion (a *plan shell*). In making this determination, the Response Taxonomy agent considers a number of factors:

- Time of attack: The timing of the response is a fundamental delineation in formulating a correct response and as such, it is the first dimension of the proposed taxonomy. The response timing may be defined as preemptive, during an attack (damage control), or after an attack (damage assessment). Preemptive responses occur when there are indications of an attack but the attack has not actually begun. Preemptive responses attempt to increase the defensive posture of the potentially affected system while continuing to provide service to users with minimal degradation of performance. Damage control responses occur when the attack has been detected and is ongoing. These responses attempt to limit the effect of the attacker while continuing to provide service to legitimate users. Damage assessment responses occur when the attack was detected after the attacker has left the system. These responses attempt to document and repair any damage to the attacked system.

- Type of attack: The type of attack is an important characterization in determining an appropriate response. For example, the response to a denial of service attack is different from a race condition attack involving a system utility. There is no attack taxonomy that is both complete (encompasses all possible attacks) and correct (appropriately characterizes attacks). The best characterization of the type of attack is the Lindqvist Intrusion Result Taxonomy [11]. It uses the CIA model as a theoretical basis for determining the type of attack and provides sufficient differentiation between the types of attacks that an appropriate response to an attack can be determined. As such, it is used as the second dimension in the proposed intrusion response taxonomy.

- Type of attacker: The type of attacker is a principal concern in determining an appropriate response. Responding to an automated attack program is different compared responding to a human attacker. Likewise, responding to a novice is quite different then effectively responding to an expert attacker. The Response Taxonomy agent classifies the attacker as either automated or human as well as either a novice or expert attacker.

- Degree of suspicion: The strength of suspicion is an important consideration in determining an appropriate response. Intrusion detection is not an exact science and as a result, intrusion detection systems can generate false positive or false negative results. The response must be tempered by the strength of suspicion that an actual intrusion is occurring. If the degree of suspicion is low, the response may be limited to account for the possibility of a false detection. If the degree of suspicion is high, a broader range of responses becomes possible.

- Attack implications: The fifth dimension of the intrusion response taxonomy is the implications of the attack. Different systems have differing degrees of importance within an organization. This difference in criticality should lead to different responses, to the same attack, against different targets. For example, the response should be different if it is a denial of service attack against a single workstation as compared to the same attack against an institutional Domain Name Server.

F. Policy Specification Agent

The Policy Specification module performs two functions: (1) it maintains any limitations on response plan steps and tactics; and, (2) it maintains the system response goal. Not all responses are appropriate in all environments. The Policy Specification agent provides a mechanism for restricting what responses are implemented in a given environment. These limitations include are legal, ethical, institutional, and resource constraints. The system administrator uses the System Administrator Interface to enter a response goal and response limitations through a checklist.

G. Tactics Agents

The Tactics agent receives an abstract plan from the Analysis agent, determines how to implement this abstract plan by mapping it into a specific set of actions, and then invokes the appropriate components of the response toolkit. For all tactics, the Tactics agent maintains an association with a set of implementations that can accomplish each tactic. These implementations are system dependent. The Tactics agent selects an implementation to execute each tactic in the plan and then invokes the appropriate component in the Response Toolkit.

In determining which implementation to select for each tactic, the Tactics agent maintains a success metric associated with each implementation. This success metric is the ratio of successful responses to an intrusion to the total

number of responses using a particular implementation. This metric is updated after each attack by the system administrator and allows the system to dynamically adjust what techniques are selected to respond to an intrusion. Those actions that are most successful are weighted so that they will be used more often than actions that are not successful. During an attack, the Tactics agent may change implementations if there are indications that the previously selected implementation is not working. These indications come principally from the Response toolkit which monitors the implementation to see if it is being executed successfully.

H. Response Toolkit

The Response Toolkit module is a collection of executables and system scripts that implement the intrusion response. These programs are system dependent and are invoked by the Tactics component. This separation of the Tactics and Response Toolkit component allows the proposed methodology to support multiple system architectures and provide a separation between the logic and implementation of the response plan.

I. User Interface

The System Administrator Interface module provides an interface for the system administrator to monitor and review incident and associated intrusion responses, suspend operation of the response system and assume an active role in the defense of the system, provide feedback to the system for adaptation, set system policy, and add new intrusion detection systems and associated interface components. The System Administrator interface receives reports from both Interface and Logger components on incidents and associated responses. These events are correlated and displayed. If the attack is ongoing, the system administrator can stop the intrusion response system and actively defend the system. After the security incident is resolved, the system administrator can indicate whether the intrusion was a real attack or a false positive report and whether the system response was successful. This allows the Interface component associated with reporting IDS to update the confidence metric associated with the IDS and the Analysis and Tactics components to update their success metrics associated with various plans and techniques. The system administrator can also set system policy through the interface. These policy specifications are recorded in the Policy Specification component and used to limit what responses the system implements. Finally, the system administrator can add new IDS and associated interface components through the System Administrator Interface component.

IV. CONCLUSIONS

This paper has proposed a methodology for adaptive intrusion response using intelligent agents. As the number and complexity of computer attacks increases, more robust intrusion response systems will be necessary. This research significantly extends previous work and provides a framework for building effective intrusion response systems.

V. REFERENCES

[1] F. B. Cohen, "Simulating Cyber Attacks, Defenses, and Consequences," *Available at http://all.net/journal/ntb/simulate/simulate.html*, May 13, 1999.

[2] C. C. Center, "CERT/CC Statistics for 1988 through 1998," *Available at http://www.cert.org/stats/cert_stats.html*, January 2000.

[3] C. C. Center, "CERT Coordination Center 1998," *Available at http://www.cert.org/annual_rpts/cert_rpt_98.html*, January 2000.

[4] C. C. Center, "Results of the Distributed-Systems Intruder Tools Workshop," *Software Engineering Institute*, Carnegie Mellon University, December 7, 1999.

[5] E. A. Fisch, "Intrusion Damage Control and Assessment: A Taxonomy and Implementation of Automated Responses to Intrusive Behavior," *Ph.D. Dissertation*, Texas A&M University, College Station, TX, 1996.

[6] G. B. White, E. A. Fisch, and U. W. Pooch, "Cooperating Security Managers: A Peer-based Intrusion Detection System," *IEEE Network*, vol. 10 (1), 1996, pp. 20-23.

[7] P. A. Porras and P. G. Neumann, "EMERALD: Event Monitoring Enabling Responses to Anomalous Live Disturbances," *Proc. 20th National Information Systems Security Conf.*, Baltimore, MD, October 7-10, 1997, pp. 353-365.

[8] P. G. Neumann and P. A. Porras, "Experience with EMERALD to Date," *1st USENIX Workshop on Intrusion Detection and Network Monitoring*, Santa Clara, CA, April 11-12, 1999, pp. http://www2.csl.sri.com/emerald/downloads.html.

[9] J. M. Bradshaw, *Software Agents*, Menlo Park, CA: AAAI Press, 1997.

[10] J. Yen and R. Lengari, *Fuzzy Logic: Intelligent, Control and Information*, New York: Prentice Hall, 1999.

[11] U. Lindqvist and E. Jonsson, "How to Systematically Classify Computer Security Intrusions," *Proc. 1997 IEEE Symp. on Security and Privacy*, Oakland, CA, May 4-7, 1997, pp. 154 - 163.