ISSN No. 0976-5697

Volume 2, No. 1, Jan-Feb 2011

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

Autonomic Intrusion Response System

G.Yamini*, S.Siva Sathya, T. Chithralekha
Department of Computer Science
Pondicherry University
Puducherry, India
{ yamini.gd, ssivasathya} @ gmail.com, tchitu@yahoo.com

R.Geetharamani
Department of Information Technology
Rajalakshmi College of Engineering
Chennai, India
rgeetha@yahoo.com

Abstract: Intrusion Response Systems (IRS) counter attacks the attempts to compromise the integrity, confidentiality or availability of a resource. Most of the IRS requires human intervention to perform their response activities, which is time consuming, whereas, Autonomic Intrusion Response Systems (AIRS) are self-healing systems inheriting their behaviour from the natural immune system. Such self-healing autonomic systems are required to safeguard the network and resources and also to provide consistent service with high quality. This paper presents the state of the art of some existing intrusion response systems along with their comparative study. It also proposes a high level architectural organization and the essential features for an ideal AIRS. It also spotlights the application of autonomic response system in agriculture –SMARTAGRI.

Keywords: Autonomous Response; Immune-Inspired System; Self-Healing; Intrusion Response; Autonomic Computing

I. INTRODUCTION

Intrusion prevention and detection tools are designed to defend networks / information systems against malicious attacks. New classes of threats with rapid speed and broad scale capabilities are emerging day to day, which attack the network. When such intrusive behavior is detected, it is preferred to perform some tasks to thwart attacks and ensure safety and availability of the network resources. Such counter-measures are called as intrusion responses [1]. To provide consistently high service quality and to ensure mission success, an autonomic self–protecting and self–healing system is required. Such systems can counter these threats with autonomic fast responses, since humans are not fast enough to react to high speed or broad scale attacks effectively.

Self-managing capabilities in a system accomplish their functions by taking an appropriate action automatically, based on one or more situations that they sense in the environment [2]. The mission of any autonomic capability is a control loop that collects details from the system and acts accordingly. Current research has been focused on better techniques for intrusion detection, whereas intrusion response is still in its infancy [3]. The Intrusion Detection System (IDS) notifies the system administrator that an intrusion is occurring or has occurred and the administrator must respond to the intrusion. Regardless of the notification mechanism employed, there is a delay between detection of a possible intrusion and its corresponding response, which provides an opportunity for attackers to exploit the resources. Using simulations, Cohen examined the effect of reaction time on the success rate of attacks [4]. The results reveal that, the success rate of the skilled attackers increases as the time delay between the intrusion detection and its response action increases. Response is a primary factor in this case [3]. If the response is autonomic and instant, then probability of a successful attack is almost negligible. Autonomic intrusion response system should lessen this delay or block the attackers until the system administrator can take an active role in defending against the attack.

The organization of the paper is as follows. Section 2 presents the state of the art of existing IRSs along with a comparative study of their capabilities. Section 3 gives an architectural overview of an AIRS. In section 4, the essential features for an efficient autonomic intrusion response system are listed. Section 5 describes about the application of autonomic response system in agriculture – SMARTAGRI. Finally, section 6 presents conclusion and future work.

II. EXISTING INTRUSION RESPONSE SYSTEMS

An autonomic intrusion response system is one that can counter the attacks on the network resources with fast responses, automatically without human intervention. The term “autonomic” is derived from human biology [2]. For example, the autonomic human nervous system automatically controls and regulates many motor and physiologic functions without conscious input from the host. The response capabilities of some of the existing intrusion response systems are briefed below and their comparisons are presented in Table II.

A. EMERALD

The Event Monitoring Enabling Responses to Anomalous Live Disturbances (EMERALD) environment is a distributed scalable tool suite [5]**.** It introduces a hierarchically layered approach to network surveillance, attack isolation, tracking malicious activity across large networks and automated response. EMERALD provides the following three services: *Service analysis* encompassing the misuse of individual components and network services within the boundary of a single domain. *Domain-wide analysis* encompassing misuse visible across multiple services and components. E*nterprise-wide analysis* encompassing coordinated misuse across multiple domains.

EMERALD has a hierarchical collection of monitors. The monitor has a defined interface for sending and receiving event data and analytical results from third-party security services. It consists of the following components: (a) Profiler Engine - Performs statistical profile-based anomaly detection, (b) Signature Engine - Performs more focused and distributed signature-analysis using a rule-coding scheme, (c) Resource Object - It is a pluggable configurable library of target-specific configuration data and methods and (d) Resolver - Performs analysis of the results produced externally by other analysis engines and implements the response polices to counter malicious activity. EMERALD provides a global detection and response capability through a recursive framework, which coordinates the dissemination of analyses from the distributed monitors. Further, EMERALD has an enhanced and versatile application programmer's interface to integrate with heterogeneous target hosts and provides a high degree of interoperability with third-party tool suites. The response actions for an intrusion [6, 7] are given in table I.

B. BlackICE

The BlackICE distributes powerful intrusion detection and protection to the entire network [8]. It provides an enterprise-wide defense web. BlackICE Sentry is a software-based intrusion-monitoring tool that observes the network for hacking activities and reports any intrusions to an ICEcap server. It employs wire-tapping technology, to collect all packets that pass by the system. BlackICE also protects telecommuting and VPN users. Depending on the purpose, it can be deployed for link monitoring or segment monitoring.

It consists of the following components: (a) BlackICE IDS Engine - Performs analyzes of incoming and outgoing network traffic in real time using the protocols that carry the information. (b) BlackICE Firewall - It provides the real-time access control. (c) BlackICE Local Console - It is the user interface which reports information about the intrusions detected, the intruders and responses graphically.

Every action reported is ranked by its severity as informational, suspicious, critical or serious. The response level indicates how BlackICE reacted when an action is detected. Low-severity actions rarely prompt a response, as they do not pose any threat to the system. Whereas, high-severity intrusions may prompt more decisive responses. The response levels are displayed as an individual icon or as an overlay on the severity icon. The response actions for an intrusion are given in table I.

Table I. Response Actions of the IRS

IRS	***Response Action***	***Description***
1. EMERALD	Generate Alarm	Alarm alerts are produced for an malicious event .
	Reconfiguration of the IDS	Reconfigures the system to collect more detailed data about the attacked hosts and networks.
	Shut down services	Stop services which are attacked.
	Isolate affected assets	The attacked resources are isolated from the network.
2. BlackICE	Visual Alarm	Flashes red (critical), orange (serious) or yellow (suspicious) light depending on the severity of the attack.
	Sound Alarm	Plays a *.wav file of our choice, whenever an attack is detected.
	Generate Evidence Files	Captures network traffic attributed to the intruder and store the detailed information in an evidence file, in the <installation directory> folder with *.enc extension.
	Traffic Filtering	Intrusion detection engine can be customized to trust specific intruders and ignore certain types of events.
3. NetProwler	Send E-mail	Sends an email message to an email recipient.
	Send an SNMP Trap	On intrusions, NetProwler acts as an SNMP Agent and sends an SNMP trap to SNMP Managers, which are configured to trap.
	Page an Administrator	Through a configured modem, it dials a paging service and pages an administrator.
	Harden a Firewall	NetProwler sends a Suspicious Activity Monitoring Protocol (SAMP) message to a configured firewall to act.
	Capture the Session	On intrusion detection, NetProwler records the remainder of the session.
	Reset Session	Terminates the session-based attack.
	Spawn a Command	Executes a specified command or batch file.
4. Cisco NetRanger	Generate Alarm	Sensor generates alarm to one or more remote Directors, where they are displayed on a graphical user interface or logged.
	Generate IP session logs	Collect information about every packet of an unauthorized user, when an intrusion has occurred for a predefined period of time.
	Reset TCP connections	After an attack, sensor can reset individual TCP connections to eliminate the threat.
	Shun the attack	Sensor has the ability to employ a network device to deny entry to an entire network.

C. Net Prowler

NetProwler [9] is a dynamic network intrusion detection system. It examines network traffic to detect, log and terminate the malicious activity in real time. It uses Stateful Dynamic Signature Inspection to detect network attacks. It has real-time signature deployment, attack signature extensibility and multi-platform host IDS integration. NetProwler resides on a dedicated server and detects network attacks without affecting the performance of other networked systems. NetProwler detects attacks by matching packet data to its database of attack signatures. It consists of the following components: (a) Agent – It is an application that scans the network segment for any intrusions like SYN flooding, TCP/IP spoofing. (b) Manager – It caches all intrusion and configuration details for the agents allocated to it. Based on the instruction from the administrator, it stores attack signature definitions and alert data, configures the agents and tune database configurations. (c) Console – It is the graphical user interface which permits the management of all agents of a particular manager, to scan the detected intrusions, assign access restrictions and monitor the state of all agents. This distributed architecture comprising of agent, manager and console provides centralized management and better network coverage.

It has an automatic application of attack signatures, which will automatically assign each new system a set of attack signatures that helps them to spot an attack. The response actions for an intrusion are given in table I. Response Actions can be configured by: (1) Individual attack signature - for each signature, the appropriate responses is assigned (2) Priority level - the actions to be taken at each high, medium and low priority are assigned.

D. Cisco NetRanger

Cisco NetRanger is a network-based intrusion detection system [10]. It uses a rule-based approach to detect intrusions. Its intrusion detection engine uses signature recognition. Context-oriented intrusion signatures, composing of network vulnerabilities, can be disclosed through the packet headers. Content-oriented signatures can be disclosed from the data fields within a packet. When the context / content of network traffic reports suspicious actions from either an authorized or unauthorized user, it automatically denies access to the intruder and reports details of the intrusion to a centralized management system. It consists of the following components: (a) Sensor - Distills large volumes of IP network traffic into meaningful security events using a rules-based engine. (b) Director - Offers a centralized graphical interface for the management of security across a distributed network. (c) Post Office- It is the backbone that allows its services and hosts to communicate with each other by a connection-based protocol. It also has a graphical user interface for managing configurations and notifying alarms. The response actions for an intrusion are given in tableI.

Table II presents a comparative study of the capabilities of the above discussed intrusion response systems.

Table II. Comparison of IRS

	EMERALD	***BlackICE***	***NetProwler***	***Cisco NetRanger***
FEATURES				
Architecture	Distributed	Centralized	Distributed	Centralized
Reusability of components	Yes	No	No	No
Interoperable	Yes	No	Yes	No
Dynamically deployable	Yes	No	Yes	Yes
Scalable	Yes	No	Yes	Yes
INTRUSION ANALYSIS				
Pattern matching of signatures	No	No	Yes	Yes
Attacks responded to	Network based, Anomaly, misuse intrusions	Host based,misuse intrusions	Network based intrusions	Network based misuse intrusions & coordinated attacks
Cost analysis of attacks	No	No	Yes	No
RESPONSE MECHANISM				
Scope of the response	Global	Local to network	Global	Local to network
Response level to new attacks	Very limited	Best	Limited	Limited
Respond to attacks in Demilitarized zone	No	Yes	Yes	No
Generate alarms	Yes	Yes	Yes	Yes
Report generation	No	Yes	Yes	Yes
Shunning the attack	No	No	No	Yes
Tracing back to the attacker	No	Yes	No	No
E-mail notification	No	No	Yes	No

III. PROPOSED ARCHITECTURAL ORGANISATION OF AIRS

Having seen the existing IRSs that are available today, this section proposes a high level architecture that shows where an AIRS fits into the intrusion detection and management environment. AIRS aims to respond to an intrusion automatically without real-time human intervention. While responding to detected intrusions, AIRS makes decision based on human knowledge and policy that is programmed into the system in advance, instead of relying on human input. The AIRS should be capable of making rational decisions that help the system meet the goals more quickly and accurately than a human could. The AIRS has the self-healing properties to recover itself from an attack. A high level architectural organization is presented in Fig. 1.

A. Managed Resources

Managed resources can be any software or hardware components. It can include resources like applications, database, networks, servers, clients, mailbox etc. These managed resources are consciously monitored by the IDS. When there are any intrusions taking place in these managed resources, it is recognized by the IDS.

B. Intrusion Detection System

The IDS continuously monitors the resources and has a set of stable conditions of the resources. It provides a standard interface through which the managed resources can be accessed and maintained. It ensures that all the managed resources are in stable operating conditions. When deviations from these stable conditions occur, the IDS capture it. It analyses the deviations detected and determines the intrusion type. The IDS has a classification set of different intrusions using which it determines the intrusion type. If the intrusion type detected is already a known type, then it directly triggers the AIRS to perform the appropriate response actions. If the intrusion type detected is a new type, then it analyses and prepares a detailed report about it. The report is then passed to the AIRS which decide on the response action to perform. The detection process also ensures that the false positive alarm rate is minimized.

C. Autonomic Intrusion Response System

The AIRS performs the self–healing task through the intelligent control loops that automates the response. The control loops are governed by policies. Using these policies, the appropriate response actions to be taken for the intrusion detected are decided. It analyses the data provided by the IDS, rapidly organizes them into sensible data and models the complex situations. This might assist the AIRS to predict future scenarios also. It determines the type of responses that has to be performed for the detected intrusion type. Then a response proposal that has to be enacted is prepared. The proposal consists of sequence of necessary changes that has to be made to the system or the managed resources. The response selected depends on various factors like intrusion severity, intruder type, time of attack, negative impact of the response and cost of the response actions. After executing the responses, its repercussions like time to heal from the intrusion, response impact on the system and response effectiveness are analyzed and updated in the policy archive. The success hit ratio is also measured for individual response action and included correspondingly in the policy archive. This ratio is given a major importance in selection of appropriate responses to the intrusion.

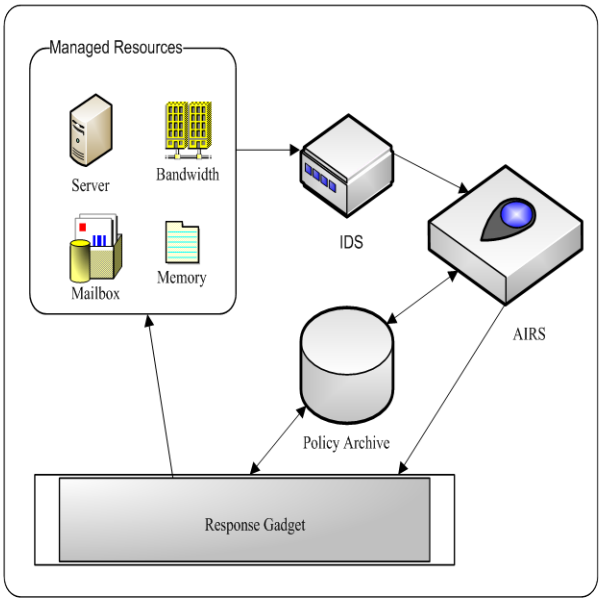

Figure 1: High-level Architecture of the AIRS

D. Response Gadget

The response proposal is executed through the response gadget. Response gadget is a library of interfaces and scripts that performs the response actions to the intrusion. These autonomic response actions performed are updated in the policy archive, which can be used for future recovery purposes. It performs the recovery functions as instructed by the AIRS on the managed resources and restores it to normal operation. The response gadget executes the self-healing operations without any human input.

E. Policy Archive

Policy Archive stores the symptoms, policies, responses, history details etc., which is used by the AIRS to develop the response proposal. It also contains the success hit ratio of the responses. These hit ratio details helps in selecting the appropriate responses from the archive. It includes a set of behavioral constraints that influences the response proposals prepared.

IV. FEATURES OF AN IDEAL AIRS

The volume and the intensity of intrusions today require rapid and automated response. Since human supervision brings in a delay into intrusion handling; the response system should have the means to counter the attacks automatically. The behavior of the existing IDS and their automated response actions are very much limited and not sufficient enough to support and counter the new stratum of threats, which are rapidly emerging every day. Thus for an effective and an efficient AIRS, some more features are needed to be supported by the response system. They are as follows:

A. Self Configuration [1, 2]

Now a days, networks have multiple components with complex interactions between them leading to dynamic environment. Intrusions in such areas cannot be well predicted. Thus the AIRS should dynamically adapt to changing environment. Self configuration to such dynamic environment will ensure continuous strength, service and productivity.

B. Continuous Self-Monitoring

Self-monitoring is a prerequisite for any intrusion handling mechanisms. It includes of a subset of awareness. AIRS should have a continuous eye on the managed resources for any intrusions. Since the system may change dynamically, only if they are continuously monitored, it can spontaneously respond to the attacks.

C. Dynamic Decision-Making and Response [11]

Once an intrusion is detected, the AIRS should automatically make decisions on the type of response that has to be taken based on the policies stored in its repository. AIRS should provide enhanced notification and recover affected resources in less time. AIRS should also take care that the future attacks of similar kind are less likely to succeed.

D. Time to Respond [12]

The time to respond is the measurement of the time the AIRS takes to perform actions against the intrusion detected. This time should be much reduced because the delay between detection of intrusion and response to that intrusion provides an opportunity for attackers to exploit the resources. The time to respond influences the sensitivity of the autonomic system to its environment.

E. Quality of Service [12]

The degree to which the AIRS attain its primary goal of responding to attacks automatically reflects its quality of service. It is highly important for an autonomic system as it improves its overall performance in terms of speed and efficiency.

F. Capability to Withstand New Attacks

AIRS, apart from responding to the attacks defined in its repository, should also be reactive to new kinds of attack. Though complete response cannot be taken against new attack, AIRS should lessen or block the attacker until the system administrator can take an active role in defending against the attack.

G. Less Human Intervention

AIRS should be able to automatically make decisions based on the policies stored and respond to attacks without much human effort. The response actions should be instantly performed in less time.

H. Cooperative [1]

When there are more than one response systems monitoring a network, then this set of response systems, collectively called cooperative response system, can combine efforts to respond to an attack. This is done globally. Performance, in terms of response speed can be better achieved in such cooperative response systems.

I. Policy-Based Adaptation Planning [13]

Policies or rules are actions that have to be taken when a particular attack occurs. Policies for different types of attacks are stored in the repository. When an attack occurs, AIRS should respond to an attack based on these policies automatically, instead of relying on human input.

J. Context Awareness [14]

AIRS should have the capability to collect information about the network in which it is present, evaluate the information and change its behavior based on the environment. Higher level of context aware adjustment enriches the realization of self-healing facet completely.

K. Global System Awareness [15]

While responding to an intrusion, AIRS decisions should not completely rely on pre-programmed responses alone, because they may be based only on local information. AIRS should consider the global state knowledge also, before taking decisions, since local reactions can impact other system components also. When selecting the response actions, it must account for the ways in which the managed resources will be affected and its resultant impact on the mission.

Thus the above features enrich the efficiency and functioning of an autonomic response system. A brief comparison of the IRS and AIRS are presented in table III. The features listed in table III can also be used as metrics for the evaluation of an autonomic response system.

Table III. Comparison between IRS and AIRS

Features	***IRS***	***AIRS***
1. Response Type	Notification System	Complete Self-Healing System
2. Human intervention to intrusion responses	Yes	No
3. Adaptively	Static	Dynamic
4. Response and recovery time	More time	Less time
5. Sensitivity	Non-adaptive system	Policy-based adaptive system
6. Fault-Tolerant	No	Yes

V. APPLICATION OF AUTONOMIC RESPONSE SYSTEM IN OTHER FIELD - SMARTAGRI

The autonomic response system can be used in the field of agriculture also. In agriculture, supply of water to the fields is a predominant and an essential activity, which can be automated. Human resources are required for this purpose. Humans need to constantly monitor the fields and supply water when required. Instead, an autonomic system can be employed for this purpose. The architectural organization of an autonomic response system for a SMARTAGRI is portrayed in Fig. 2. The components of it are (a) Water-Level Proctor, (b) Analyzer, (c) Response Gadget and (d) Water pump. The working of SMARTAGRI is described below:

A. Water-Level Proctor

Water-level proctor consistently checks and measures the water-level content ($Water_{Level}$) of the field. If the water content of the field is within the normal (expected) range ($Water_{Norm}$), then no response action (ie. water supply) is needed for the field at that moment. When the water-level content is below the normal range, water supply is needed for the field and so it triggers the Analyser.

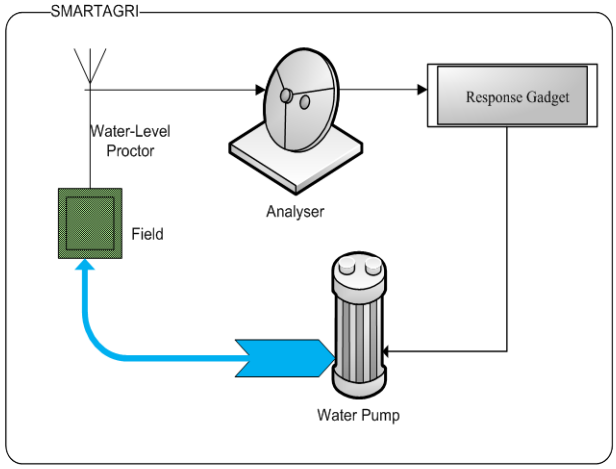

Figure 2: Architectural Organization of SMARTAGRI

B. Analyzer

Analyzer receives the $Water_{Level}$ value from the proctor. Based on that value, it decides the amount of water supply ($Water_{Amt}$) required for the field and activates the Response Gadget.

$$Water_{Amt} = Water_{Norm} - Water_{Level} \quad (1)$$

$Water_{Amt}$ value gives the exact amount of water required by the field thus averting excess water supply to the field.

C. Response Gadget

Response Gadget, on receiving the $Water_{Amt}$ value, sets the water outlet value on the water pump accordingly and triggers it to on.

D. Water Pump

Water Pump pumps and supplies water to the field according to the set $Water_{Amt}$ value.

As water is supplied to the fields based on the $Water_{Amt}$ value, SMARTAGRI also helps in the conservation of water by enriching the fields with required adequate water supply only. Thus an autonomic response system holds its flag up in other fields too.

VI. CONCLUSION

Development of self-healing autonomic intrusion response system is a challenging field for the researchers. Although complete automatic self-healing may be difficult to achieve due to the presence of novel intrusions, significant reduction of human effort is desirable. This paper presented a survey of some existing intrusion response systems along with a comparative study of their capabilities. Proposed high level architectural description of an AIRS and the features required for an ideal AIRS were also stated. The adaptiveness of the autonomic response system is highlighted through its application in other fields like in agriculture – SMARTAGRI. Our future work resides on the development of a complete, detailed architecture for an AIRS, possessing the features listed in this paper.

VII. ACKNOWLEDGMENT

This work is a part of the AICTE funded project titled ‘Bio-inspired Intrusion Response System through feature relevance Analysis on Attack Classification’, Under the Research Promotion Scheme (RPS), Ref. No: 8023/BOR/RID/RPS-59/2009-10.

VIII.REFERENCES

[1] N. Stakhanova, S. Basu and J. Wong, “A taxonomy of intrusion response systems,” International Journal of Information and Computer Security, vol. 1, ACM, Jan 2007, pp.169 - 184.

[2] IBMGroup, An architectural blueprint for autonomic computing,http://www.ginkgo-networks.com/IMG/pdf/AC_Blueprint_White_Paper_V7.pdf, June2005.

[3] A. Curti and J. Carver, "Intrusion Response Systems: A Survey," Department of Computer Science, Texas A&M University, 2000, Tech Report.

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

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

[6] P. G. Neumann and P. A. Porras, “Experience with Detection EMERALD to Date,” 1st USENIX Workshop on Intrusion and Network Monitoring, Santa Clara, CA, April 11-12, 1999, http://www.usenix.org/event/detection99/full_papers/neumann/neumann.pdf.

[7] M. Crosbie and K. Price, COAST Laboratory, Purdue University, Intrusion Detection Systems, http://www.cerias.purdue.edu/about/history/coast_resources/idcontent/ids.html.

[8] Network Ice, BlackIce User's Manual, version 1.0, http://documents.iss.net/literature/ICEcap/BlackICE_Sentry_User_Guide30.pdf, Jan. 26, 2001.

[9] Axent Technologies, NetProwler User Manual Version3.0, http://www.neocom.pl/dokumenty/Axent/np303.pdf, May1999.

[10] Cisco Systems, Net Ranger User Guide, http://www.dowers.net/ftp/TechnicalDocs/NetRangerOverview.pdf.

[11] B.Foo, M.W. Glause, G.M. Howard, Y. Wu, S.Bagchi and E.H.Spafford, “Intrusion Response Systems: A Survey,” Center for Education and Research Information Assurance and Security, Purdue University, Tech Report 2008-4.

[12] McCann J .A. and Huebscher M.C, “Evaluation Issues in Autonomic Computing,” International Workshop on Agents and Autonomic Computing and Grid Enabled Virtual Organizations (AAC-GEVO’04), 3rd International Conference on Grid and Cooperative Computing Wuhan, China, 21-24 October 2004.Springer-Verlag, Heidelberg.

[13] McCann JA and Huebscher M., **"**A survey of Autonomic Computing - degrees, models and applications**,**" ACM Computing Surveys, ACM, 2008.

[14] C. Klein, R. Schmid, C. Leuxner, W. Sitou and B. Spanfelner, "A Survey of Context Adaptation in Autonomic Computing," Proceedings of the Fourth International Conference on Autonomic and Autonomous Systems, 2008 (ICAS 2008), IEEE Computer Society 2008, pp. 106 – 111.

[15] S.M.Lewandowski, D.J.V.Hook, G.C.O'Leary, J.W.Haines and L.M.Rossey, "SARA: Survivable Autonomic Response Architecture," Proceedings of the DARPA Information Survivability Conference and Exposition II, 2001, IEEE Computer Society, Anaheim, CA, USA, vol. 1, pp. 77 – 88.