International Journal of Scientific Research in Computer Science and Engineering
Vol.5, Issue.5, pp.22-34, October (2017)

Survey Paper

E-ISSN: 2320-7639

A Security Determination-Reaction Architecture for Heterogeneous Distributed Network

B.Bhasker[1*], T. Jagadish kumar[2], M.V.Kamal[3]

[1*]Department of CSE, Malla Reddy College of Engineering and Technology, Hyderabad, India
[2]Department of CSE, Malla Reddy College of Engineering and Technology, Hyderabad, India
[3]Department of CSE, Malla Reddy College of Engineering and Technology, Hyderabad, India

**Corresponding Author: bhasker.b90@gmail.com Tel: +91-70935 90361*

Available online at: www.isroset.org

Received 12th Sep 2017, Revised 26th Sep 2017, Accepted 17th Oct 2017, Online 30th Oct 2017

Abstract— The main focus of this paper is to provide a global architectural solution built on the requirements for a reaction after alert detection mechanisms in the frame of Information Systems Security and more particularly applied to telecom infrastructures security. These infrastructures are distributed in nature, therefore the targeted architecture is developed in a distributed perspective and the architecture is elaborated using the multi-agent system. The Multi-Agent System decision-reaction architecture is developed in a distributed perspective and is composed of three basic layers: low level, intermediate level and high level. The low level aim to be the interface between the main architecture and the targeted infrastructure. The intermediate level is responsible of correlating the alerts coming from different domains of the infrastructure and to deploy smartly the reaction actions.This intermediate level is elaborated using multi-agents system that provide the advantages of autonomous and interaction facilities.The high level permits to have a supervision view of the whole infrastructure, and to manage business policy definition.The proposed approach has been successfully experimented for data access control mechanism.The proposed approach has been illustrated based on the network architecture for heterogeneous mobile computing developed by the BARWAN project .Accordingly: The Building Area constitutes the low level. The Campus –Area is the intermediate level. It takes care about the alerts coming from different domains and deploy the reaction actions smartly.The multi-agent system that has been associated to the OntoBayes model for decision support mechanism.This model helps agents to make decisions according to preference values and is built upon ontology based knowledge sharing , bayesian networks based uncertainity management and influence diagram based decision support.

Keywords – Security Policy; Multi-agents systems Architecture; decision system; reaction; Distributed networks; bayesian network

I. INTRODUCTION

Today information systems and mobile computing networks are more widely spread and mainly heterogeneous. This basically involves more complexity through their opening, their interconnection, and their ability to make decisions. Consequently, this has a dramatic drawback regarding threats that could occur on such networks via dangerous attacks .This continuously growing amount of carry out malicious acts encompasses new and always more sophisticated attack techniques, which are actually exposing operators as well as the end user. State of the art in terms of security reaction is limited to products that detect attacks and correlate them with a vulnerability database but none of these products are built to ensure a proper reaction to attacks in order to avoid their propagation and/or to help an administrator deploy the appropriate reactions.

In the same way, says that at the individual host-level, intrusion response often includes security policy reconfiguration to reduce the risk of further penetrations but doesn't propose. another solution in term of automatic response and reaction. It is the case of CISCO based IDS material providing mechanisms to select and implement reaction decision.

The realm of security management of information and communication systems is actually facing many challenges due to the fact that it is very often difficult to:

- Establish central or local permanent decision capabilities
- Have the necessary level of information
- Quickly collect the information, which is critical in case of an attack on a critical system node
- Launch automated counter measures to quickly block a detected attack .

Information security management and communication systems is actually in front of many challenges due to the fact that it is very often difficult to establish central or local permanent decision capabilities, have the necessary level of

information, quickly collect the information, which is critical in case of an attack on a critical system node, or launch automated counter measures to quickly block a detected attack.

Based on that statements, it appears crucial to elaborate a strategy of reaction after detection against these attacks Our previous work around that topic has provided first issues regarding that finding and has been somewhat presented in and.
These papers have proposed an architecture to highlight the concepts aiming at fulfilling the mission of optimizing security and protection of communication and information systems which purpose was to achieve the following:

- Reacting quickly and efficiently to any simple attack but also to any complex and distributed ones;
- Ensuring homogeneous and smart communication system configuration, that are commonly considered and the main sources of vulnerabilities.

One of the main aspects in the reaction strategy consists of automating and adapting policies when an attack occurs. In scientific literature a large number of definitions for policy and conceptual model exist. The most famous are Ponder and Ponder2 , Policy Description Language and Security Policy Language . For the purpose of that paper, we prefer the one provided by Damianouetal . in : Policies are rules that govern the behavior of a system**.**

The provided policy adaptation is considered as a regulation process. The main steps of the policy regulation are described in Fig. 1, which shows the process that takes the business rules as input, and maps them onto technical policies. These technical policies are deployed and instantiated on the infrastructure in order to have a new state of temporary network security stability adapted to the ongoing attack.

This policy regulation is there after achieved in modifying or adding new policy rules to reach a new standing (at least up to the next network disruption) policy based on the observation of the system's current situation. It must be specified that this regulation process rely also on policies adaptation to a specific context. Those contexts and the modeling of concepts of org, role, activity, view are explained in Efficiently react against an attack, especially if this needs a change on an equipment configuration, often necessitates many checks that have to be performed in order to avoid bad side effects (conflict creation, services stability, etc.).

In this paper, we focus our work on policy deployment and on policy modification decision-reaction challenges as highlighted in the rounded rectangle of Fig. 1. This two fold challenge has already been addressed by other researches likein. Torrellas explains that facilitating timely decision-making may achieve much greater productivity benefits by engineering network security systems using multi-agents.
In , You developed the concepts of teleservice and proposed an implementation of an e-maintenance platform based on a Mulit-Agent System (MAS). You explained how a Case-Based reasoning method may be used to improve the autonomous decision-making ability.
Others' works propose rather similar solutions like but none are explicitly dedicated to the management of security alerts reaction in the field of open and heterogeneous networks.
Consequently, the combination of the reaction mechanism with the decision support system remains, for those solutions, a poorly addressed requirement in parallel to other more specific constraints related to the characteristics of the context.

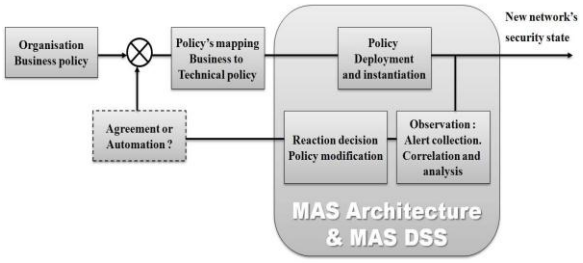


Figure 1.Policy regulation

To illustrate this decision mechanism, we use the results of the BARWAN1 project. This project focused on enabling truly useful mobile networking across an extremely wide variety of real-world networks and mobile devices. The case study analyzed by the project is a medical application enabled by wide-area wireless and that exploits the Berkeley InfoPad pooled computing power to permit a small number of workstations to support a large number of end users. Fig.3 highlights the distribution of the application over the buildings, the campus and the metropolitan layers. In that paper, an architecture is proposed to adapt a reaction once an attack occur on one of those layers.
Additionally, the architecture makes is possible to integrate internal or external contextual information for the reaction decision like, i.e. the usage of the application, as proposed in the case study, during a medical rescue operation after a serious auto accident on Golden Gate Bridge2 .
The next section introduces the MAS architecture, section 3 exposes the combination decision support system as well as its with the MAS, and the last section concludes the paper.

Consequently, policy regulation's automation needs in one hand the existence of a hierarchy between the rules in case of multiple choices due to multiple attacks, and in second hand an automatic method to validate the policy's modifications. At the business level, the targeted foreseen solution will be able to improve the resilience to attacks of core IP networks and, by extension to large information

systems, which form critical infrastructures for communication and services today. The second section of this paper introduced requirement that has to be taken into account for the definition the presented architecture and introduce agent based policy management architecture.

II. REQUIREMENTS ANALYSIS AND DESIGN

The architecture of such a reaction system must respect some classes of requirements that has been synthesized in the following:

Business needs: Laws and regulations dedicated to private sector exist and are continuously improving requirements that enforce the top management to be responsible regarding the needs for information security (SOX, Basel 2, ISO27000).Corporate policy and security policies are tools under the cover of the business that face IS security issues. In that sense, security requirements are dictated by the business and IT staff implements them. Accordingly, a business requirement is: when an attack occurs, the technical IT committee adapts the basic policy to solve the problem.

Scalability: The system should be able to manage and ensure security of several sub-systems (e.g. LAN and subs-LAN) called "managed systems".

Availability: There's always in IT systems a single element, component, system, device, or person that is crucial for the mission and ofcourse the security; these item are called "single points of failure" and the management system should avoid them.

Confidence: Current usage of automatic reaction technologies is narrowed by end-user confidence into the system. As a result,operators often deactivate automatic features of the system.Strong confidence can be established by design, ensuring that reaction don't contravene known business policies

Autonomy: However, certain autonomy should be provided to the managed systems, to avoid paralyzing situation in case of loss of connection with the global system.

Survivability and robustness:
The management system should implements means for being able to continue to function during and after a damage or loss due to intentional malicious threats (i.e. survivability) ,and unintentional hardware failures, human errors, etc. (e.g. robustness).

Reaction applicability:
A reaction should be applicable to several managed systems or to targeted objects. The reaction applicability should be specified and adaptable considering the reaction. Furthermore, a time defining the validity of the reaction should be specified (temporary reactions for a certain time, or permanent).

Alert management correlation:
Relatively to the alerts management, a global correlation between the alerts coming from different managed systems should be realized. The existing intrusion detection tools generate alerts and the system just collect and process them, as observation input.

Global supervision: Furthermore, a global supervision must available in order to manage detection and reaction on widely spread systems. Indeed, alerts from all the managed systems should be correlated. together at the higher level of hierarchy.

III. MULTI AGENT SYSTEM ARCHITECTURE

A. Overview and Definitions:
A Multi-Agent System (MAS) is a system composed of several agents, capable of mutual interaction. The interaction can be in the form of message passing or producing changes in their common environment.
Agents are pro-active, reactive and social autonomous entities able to exhibit organized activity, in order to meet their design objectives, by eventually interacting with users. Agent is collaborative by being able to commit itself to the society or/and another agent.

- An agent encapsulates a state and a behavior and provides moreover a number of facilities that are:
- An agent has control on its behavior
- An agent decides in which state it is, even if external event may influence this decision.
- An agent exerts this control in various manners (reactive, directed by goals, social)
- MAS have several control flows while a system with objects has a priori only one control flow.
- The agents also have global behavior into the MAS, such as:
- Cooperation: agents share the same goal
- Collaboration: agents share intermittently the same goal,
- Competition: incompatible goals between agents.

An architecture description has been developed considering the requirements described in the previous section. To manage several different systems, due to their location, the focused business domain or organization type, a distributed system is appropriate. Furthermore, a distributed solution should be able to bring some autonomy to the managed systems; robustness, survivability and availability are also impacted. The architecture will be composed of several components, called "nodes or operators", having different responsibilities. Theses nodes will be organized in two dimensions, as presented in Figure 2.

OrBAC Model :

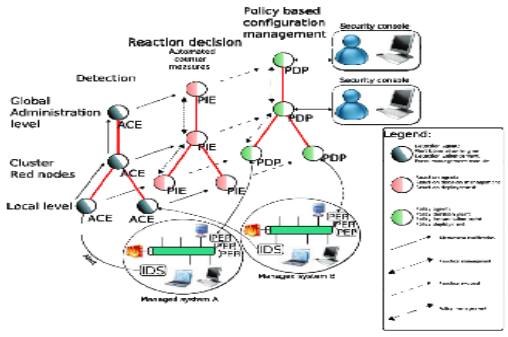


Figure 2. Reaction Architecture Overview

The vertical dimension, structured in layers relatively to the managed network organization, allows adding abstraction in going upward. Indeed, the lowest layer will be close to the managed system and thus being the interface between the targeted network and the management system. The higher layer will expose a global view of the whole system and will be able to take some decisions based on a more complete knowledge of the system, business, and organization.

Intermediate levels (1 to n-1) will guarantee flexibility and scalability to the architecture in order to consider management constraints of the targeted infrastructure. Those middleware levels are optional but allow the system to be better adapted to the complexity of a given organization and the size of the information system. The horizontal dimension, containing three basic components, is presented in Fig. 2 and its three main phases are described below:

1) Alert: Collect, normalize, correlate, analyze the alerts coming from the managed networks and represent an intrusion or an attack. If the alert is confirmed and coherent, it is forwarded to the reaction decision component. (Alert Correlation Engine-ACE).

2) Reaction Decision: Receive confirmed alerts for which a reaction is expected. Considering the knowledge of: policy, the systems' organization and specified behavior, these components decide if a reaction is needed or not and define the reaction, if there is any. The reaction will be modification(s), addition(s) or removal(s) of current policy rules. (Police Instantiation Engine-PIE).

3) Reaction: Instantiation and deployment of the new policies, on the targeted networks. The deployment (Policy Deployment Point – PDP) and enforcement (Policy Enforcement Point – PEP) of these new policies, lead to a new security state of the network. The terminology in italic used in section 4 is extracted from both: XACML and OrBAC Model .

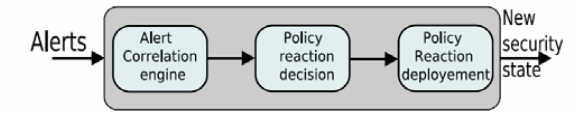


Figure 3. The three basic components

An issue is raised considering which layer is allowed to take a decision reaction: only one layer, two, several, or all? If more than one layer can trigger a reaction on the same object(s), there will be a conflict issue. Thus, the system should be able to provide mechanisms to solve conflicts between several selected reactions. Another issue concerns the agreement: at which level should it be asked? A solution could be to ask at the same level (or at an upper one) that the reaction decision is made; this should be specified by the user. A possible solution is a distributed, vertically layered and hierarchical architecture. The layer's number could be adapted according to the managed systems' organization. In our case, three layers are sufficient (local, intermediate and global). The reaction system is composed of three main parts: the alert management part, the reaction part and the police definition-deployment part. Three trees (alert, reaction and policy) could be placed side by side, as presented in Fig 2. Fig 2. explains how the reaction architecture is mapped onto the BARWAN network (borrowed from). The three layers are from top to bottom: The metropolitan Area,The campus area, and the in-building network (building A and B). The next step of our research development is firstly the definition of a reaction engine that encompasses both, architecture components and the communication engine between these components. This engine is based on a message format and on a message exchange protocol based on standards such as . Secondly, real cases are studied in order to experiment with the architecture and its associated protocol.

Response time=2*(propagation time between levels)
+ (processing and deployment time)

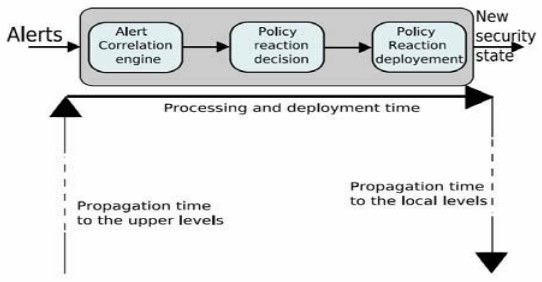


Figure 4. Response time

The next step of our research development is firstly the definition of a reaction engine that encompass both architecture components defined in that paper and communication engine between these components. This engine will be based on a message format and on a message

exchange protocol based on standards such as . Secondly, real cases must be studied in order to experiment with the architecture and its associated protocol.

The message format is defined in XML format and is structured around a number of attributes that specify the message source, the message destination and the message type (alert, reaction, policy request, policy modification, policy modification validation, decision and synchronization). The protocol defines the exchange format and the workflow of messages between the architecture components. It encompasses a set a rules governing the syntax, semantics, and synchronization of communication. The technical requirements request the operator structure must be flexible in order to be able to reorganize itself, if an operator fails or disappears. Each operator also has to be autonomous in order to permit reorganization. Given these requirements, the use of MAS appears as a solution to provide autonomy, flexibility and decision mechanisms to each operator that are consequently represented by agents.

As studied in the state of the art presented in , a set of agents could be managed and controlled through an organization. An organization is a set of agents playing roles, gathered in a normative structure and expecting to achieve some global and local objectives. Several models like the roles model, the tasks model, the interaction model or the norms models specify an organization.

In our context we need an interaction definition in order to specify communication protocols between agents representing operators. We also need roles in order to specify which agent will have to communicate or act in order to detect intrusions and then react. Based on this needs, the use of an electronic institution based on agents is one of the possibilities that we will investigate.

The main goal of the reaction policy enforcement engine is to apply policies in terms of specific concrete rules on "technical" devices (firewall, fileserver, and other systems named PEP). For that, we need means to make ACE, PIE, PDP and PEP interact and collaborate. Fig 5.

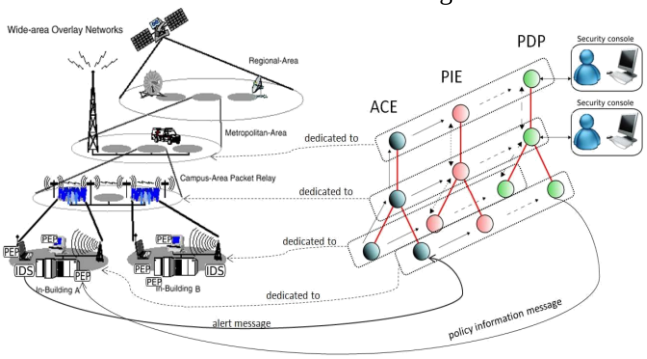


Figure 5.Mapping of the BARWAN architecture with Multi-Agent System reaction architecture

Fig. 5 introduces the developed architecture. The flow is supposed to begin with an alert detected by the IDS positioned on the InfoPad server. This alert is send to the Building A_ ACE agent. This ACE agent confirms or not the alert to the PIE. This decision to confirm the alert is explained in section 3. Afterwards, the PIE decides to apply new policies or to forward the alert to an ACE from a higher layer (upper ACE). Its PIE agent sends the policies to the PDP agent, which decides which PEP is able to implement it in terms of rules or script on devices (Info Pad server, fileserver, etc.) Then, the PDP agent sends the new policy to the Info Pad PEP agent that knows how to transform a policy into a rule or script understandable by the Info Pad server.

On Fig. 5, dash dot lines stand for flow of messages encompassing alert or alert confirmation. Full lines stand for flow of messages containing policies information, and dot lines are reserved for decision support mechanisms. The following sections present the specification of the policy enforcement engine deployment based on agents. After motivating this solution, we introduce agents and multi-agents theory and we detail the Policy Enforcement Point, Policy Decision Point and the communications between them.

The multi-agents systems concept already defines architectures and models for autonomous agents' organization and interaction. Existing platforms like JADE (Java Agent Development Framework) implement agents' concepts as well as their ability to communicate by exchanging messages and the reaction components integration could be simplified. This is a solution, which will be detailed here after. The Foundation for Intelligent Physical Agents (FIPA) promotes the success of emerging agent based applications, services and equipment. It makes available internationally agreed specifications that maximize interoperability across agent based applications; services and equipment pursue this goal. This is realized through open international collaboration of member organizations, which are companies and universities active in the agent field. FIPA's specifications are publicly available. They are not technologies for specific application, but generic technologies for different application areas, and not just independent technologies but a set of basic technologies that can be integrated by developers to make complex systems with a high degree of interoperability.

The used multi-agent framework is JADE. We ase ourselves on a survey made in to argue that this agent platform responds to the expectations in terms of agents' functionalities, security, performance, standardization, and secure communication between agents.

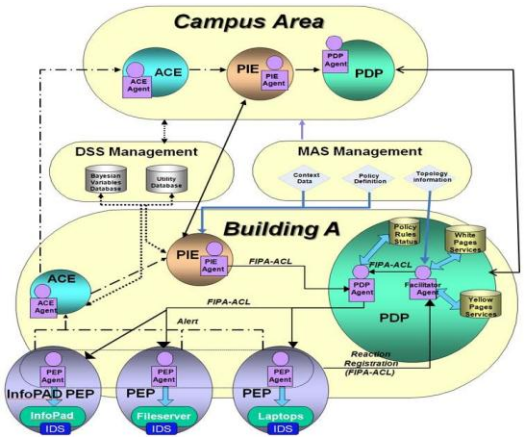


Figure 6. Multi-Agent System reaction architecture

A focused analysis of the PDP shows that it is composed by several modules. For the multi-agent system point of view, the Component Configuration Mapper results from the interaction between the PDP agent and the Facilitator Agent while the Policy Analysis module is realized by the PDP agent. The Facilitator manages the network topology by retrieving PEP agents according to their localization (devices registered with IP address or MAC address) or according to actions they could apply and their type (firewall, file server, etc.). For that the Facilitator uses white pages and yellow pages services. The JADE platform already provides implemented facilitator and searching services. Besides, the use of a multi-agent system as the framework provides flexibility, openness and heterogeneity. Actually, when we decide to add a new PEP, we just have to provide its PEP Agent with the ability to concretely apply the policies that will register itself through the Facilitator, which will update the databases.

IV. DECISION SUPPORT ARCHITECTURE

Section 3 explains the developed MAS architecture that guarantees a telecommunication security incident reaction. Section 4 explains the implementation of the decision mechanism. The MAS architecture has voluntarily been explained before the Decision Support System (DSS) part because components of this architecture are used for the illustration of the DSS.

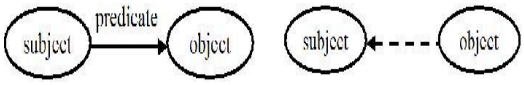
One important challenge of the DSS is the management of uncertainty. Uncertainty is defined as situation "caused by a lack of knowledge about the environment when agents need to decide the truth of statement*."*

Decision is a process and consequently, it may be represented using its input and its output. For the security incident reaction, inputs of the alert sending decision mechanism are for instance: the severity, duration and frequency of the alerts, the contribution of the system to the medical rescue operation (if any), or the criticality of that rescue operation. Outputs of the process are for instance: the escalation of the alert to upper ACE or its confirmation to the PIE. For the clarity of the paper, some parameters from the case study will be partially omitted.

As explained by Yang, the decision-making mechanism is composed of four pillars: Ontology, Bayesian Networks (BN), Influence Diagram (ID) and Virtual Knowledge Community (VKC). In the framework of that paper, the VKC will not be treated because the use of the 3 first pillars is enough to understand the decision mechanism. The approach preferred to design the decision mechanism is adapted from the research performed by Yang's thesis for the Incident reaction through a MAS architecture. As a consequence our solution differs from and completes the architecture for incident reaction that is really deployed in our research labs.

A. **Ontology** Ontology is the first pillar and is defined by *a* formal, explicit specification of a shared conceptualization . Ontology may be categorized as domain ontology when it concerns concepts and their relations from a same and well defined domain or top-level ontology when it concerns very general domain-independent concepts. Ontology is the most important pillar in that, it will be adapted to support the second pillar concerning the Bayesian Network and the third pillar concerning the Influence Diagram.

For the incident reaction system, ontology is defined using the Web Ontology Language (OWL). Resource Development Frameworks (RDF) syntax is the most commonly used method to model information or meta concepts in OWL. It may be implemented in web resources and is structured based on the triple (object, subject, predicate). Fig. 5 illustrates RDF graph. Both, object and subject are resources whereas predicate is an attribute or a relation used to describe a resource.



In parallel to the MAS architecture developed in section 4, we need a DSS to decide the transfer of an alert from the IDS to the Building A ACE3, for the forward of that alert to an upper ACE, and for the confirmation of the alert to the PIE. This is formalized using OWL as explained in Fig. 8. On that figure, ovals stand for OWL class, solid arrow lines stand for RDF predicate, dash arrows for influence relations and rounded rectangles for set of domain value.

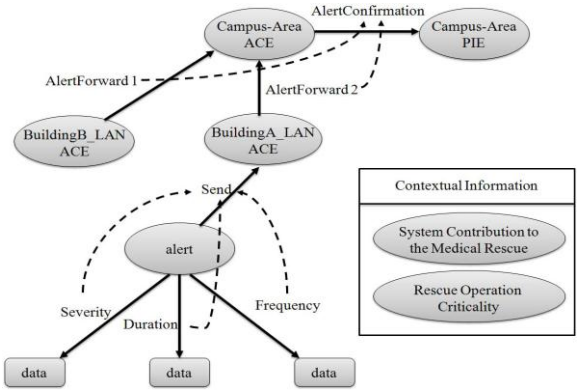


Figure 8. Decision system for alert transfer using OWL

B. Onto Bayes Ontology developed in the previous section permits to formalize the concept encompassed in the MAS architecture as well as their relations. However, at that the ontological level of formalization, uncertainty challenge remains unaddressed and decision mechanism remained needed for the agents to take the decision. Onto Bayes is an extension of OWL with two features: Bayesian Networks and Influence Diagram. BN address the uncertainty and ID support the decision mechanism process.

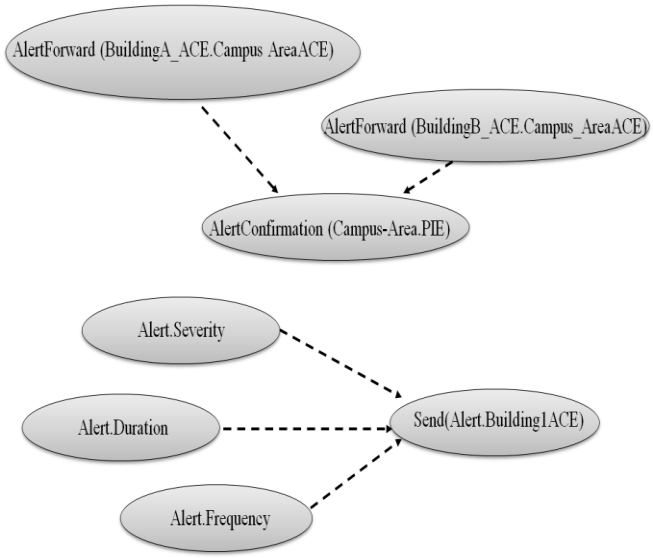


Figure 9. Bayesian graph models for alert sending and alert confirmation processes

The ovals represent Bayesian variables and the arrows specify their relations. The graph is to be read i.e.

1. The alert that is forwarded from the Building B ACE to the network upper ACE has influence on the confirmation of the alert that is send from the Campus-Area ACE to the PIE. I.e.

2. The severity of the alert has influence on the action to send an alert to the Building A ACE. The last examples maybe translated using the new OWL depends On element as following :

```
<owl:Class rdf:ID=”alert.severity”>,
<owl:Restriction>
<owl:onProperty>
<owl:ObjectProperty red:ID=”dependsOn”/>
</owl:onProperty>
<owl:hasValue rdf:resource=”system.impact”>
</owl:Restriction>
</owl:class>
```

Figure 10. Dependency encoding

BAYESIAN NETWORK:

A Bayesian network, Bayes network, belief network, Bayes model or probabilistic directed acyclic graphical model is a probabilistic graphical model that represents a set of random variables and their conditional dependencies via a directed acyclic graph (DAG). For example, a Bayesian network could represent the probabilistic relationships between diseases and symptoms. Given symptoms, the network can be used to compute the probabilities of the presence of various diseases.

TABLE I : BAYESIAN VARIABLES VALUE PROBABILITY

Probce ll	Has p parameters	Has p value
Cell 1	alert.severity=low|rescue.impact=low	0.6
Cell 2	alert.severity=medium|rescue.impact=low	0.3
Cell 3	alert.severity=high|rescue.impact=low	0.1
Cell 4	alert.severity=low|rescue.impact=medium	0.2
Cell 5	alert.severity=medium|rescue.impact=medi um	0.5
Cell 6	alert.severity=high|rescue.impact=medium	0.3
Cell 7	alert.severity=low|rescue.impact=high	0.1
Cell 8	alert.severity=medium|rescue.impact=high	0.2
Cell 9	alert.severity=high|rescue.impact=hig	0.7

The quantitative extension is performed with the association of probability table to the Bayesian variables. In the case of the BARWAN example, the Table 1 provides de quantitative probability P and is represented on Fig. 2 by the Bayesian variables database.

The conditional probability from Table I is encoded as follows (Fig. 11):

```
<owl:Class rdf:ID=”Alert”>
<CondProbDist rdf:ID=”table_1”>
<hasPCell>
<Prob C rdf:ID=”Cell_1”>
<HasPValue rdf:Id data type=”#float”>
>0,8</HasPValue>
<HasParameters rdf:data type=”#string”
>alert.severity=low|rescue.impact=low>
</HasParameters>
</ProbC>
</HasPCell>
...
</CondProbDist>
</owl:Class>
```

Figure 11. Bayesian variables value probability encoding

V. CONCLUSIONS

In this paper we have presented a global and integrated decision-reaction architecture developed for an incident reaction system and based on a policy regulation approach strategy. The main advantage of this architecture is its distributed structure. Moreover, the architecture covers the requirements needs described in section II. The solution is composed firstly with a MAS . MAS react quickly and efficiently against an attack while being adapted for heterogeneous and distributed networks and secondly with a decision support system that helps agents to make decisions based on utility preference values. This is achieved by taking uncertainty into account through Bayesian networks and influence diagram. The architecture has been illustrated based on the network architecture for heterogeneous mobile computing developed by the BARWAN project. Accordingly, contextual information has been introduced in the decision mechanism like i.e. the criticality of the medical rescue operations. The decision support system has been explained for the transfer of an alert from the alert correlation engine to the policy instantiation engine. Other decision points exist in the architecture. All of them could be solved using decision support system. The future works based on our achievements will be the specification of a protocol, specification of the messages and thus the reaction methodology service oriented based. This protocol and methodology will be dedicated to the architecture presented in this paper and address the interoperability issues with regard to the policy representation and modeling.

VI. ACKNOWLEDGEMENT

This research was funded by the National Research Fund of Luxemburg in the context of SIM (Secure Identity Management - FNR/04/01/03) and TITAN (Trust-Assurance for Critical Infrastructures in Multi-Agents Environments, FNR CO/08/IS/21) project.

REFERENCES

[1] A. Cuevas, P. Serrano, J. I. Moreno, C. J. Bernardos, J. Jähnert, R. L. Aguiar, V. Marques, *Usability and Evaluation of a Deployed 4G Network Prototype*, Journal of Communications and Networks, Vol. 7 (2), 2008.

[2] Teo, Joseph Chee Ming; Tan, Chik How; Ng, Jim Mee, *Denial-ofservice attack resilience dynamic group key agreement for heterogeneous networks*, Telecommun. Syst. 35, No. 3-4, 141-160 (2007).

[3] L. J. LaPadula. *State of the Art in Anomaly Detection and Reaction Technical Report MP 99B0000020*, Mitre, July 1999.

[4] G.L.F. Santos, Z. Abdelouahab, R.A. Dias, C.F.L. Lima, E. Nascimento , E.M. Cochra. *An Automated Response Approach for Intrusion Detection Security Enhancement*, Software Engineering and Applications, 2003.

[5] M. Petkac and L. Badger, *Security agility in response to intrusion detection* in 16th Annual Conference on Computer Security Applications (ACSAC '00), 2000.

[6] C. Feltus, D. Khadraoui, B. de Rémont and A.Rifaut, *Business Gouvernance based Policy regulation for Security Incident Response.* IEEE Global Infrastructure Symposium, 6 July 2007.

[7] Gateau, D. Khadraoui, C. Feltus, *Multi-Agents System Service based Platform in Telecommunication Security Incident Reaction*, IEEE Global Information Infrastructure Symposium, 2009.

[8] N. Damianou, N. Dulay, E. Lupu, M. Sloman , *The Ponder Policy Specification Language, Workshop on Policies for Distributed Systems and Networks (Policy2001*), HP Labs Bristol, 29-31. Springer-Verlag.

[9] Bertino, E., Mileo, A., and Provetti, A. 2005. *PDL with Preferences. IEEE international Workshop on Policies For Distributed Systems and Networks, Policy 2005 – Vol. 00*, IEEE Computer Society, Washington, DC, 213-222.

[10] Aamodt, A., Plaza, E., 1994. *Case-based reasoning: foundational issues, methodological variations, and system approaches.* AI Communications IOS Press 7 (1), 39–59.

[11] K.-Y. Lu, C.-C. Sy, *A real-time decision-making of maintenance using fuzzy agent*, Expert Systems with Applications, Volume 36, Issue 2, Part 2, March 2009, Pages 2691-2698

[12] Carrascosa et al., 2006 C. Carrascosa, J. Bajo, V. Julian, J.M. Corchado and V. Botti, *Hybrid multi-agent architecture as a real-time problem-solving model*, Expert Systems with Applications 34 (2006), pp. 2–17.

[13] Basile, C.; Lioy, A.; Perez, G. Martinez; C., F. J. Garcia; Skarmeta, A. F. Gomez, *POSITIF: A Policy-Based Security Management System*, Policies for Distributed Systems and Networks, 2007. POLICY’07, pp. 280 – 280.

[14] Torrellas, G.A.S, *Modelling a network security systems using multiagents systems engineering*, IEEE International Conference on Systems, Man and Cybernetics, 2003. Vol 5, (5-8). 2003 pp 4268 - 4273.

[15] R. Yu, B. Iung, H. Panetto, *A multi-agents based E-maintenance system with case-based reasoning decision support*, Engineering Applications of Artificial Intelligence, Vol. 16, Issue 4, June 2003, Pages 321-333

[16] http://xml.coverpages.org/draft-seitz-netconf-xacml-00.txt

[17] Cuppens, F., Cuppens-Boulahia, N., Miège, A.: *Inheritance hierarchies in the Or-BAC Model ad application in a network environment. In:* Second Foundations of Computer Security Workshop (FCS’04), Turku, Finland (2004).

[18] F. Cuppens and A. Miège, *Modelling contexts in the Or-BAC model, 19th Annual Computer Security Applications Conference*, Las Vegas, December, 2003
[19] *IDMEF/RFC4765, Network Working Group: Hervé Debar, France Telecom*; D. Curry, Guardian; B. Feinstein, SecureWorks, Inc.; March 2007
[20] B. Gâteau. Modélisation et Supervision *d'Institutions Multi-Agents*. Ph.D. Thesis, Ecole Supérieure des Mines de Saint-Etienne, 2007.

Author's Profile

Mr.B.Bhasker, Completed Bachelor of Technology from JNTU Hyderabad University in the year 2012 from Information Technology, did his M.Tech-CSE in 2014 from JNTUH. Presently working as Assistant Professor in CSE Dept of Malla Reddy Group of Institutions located in Hyderabad, Telangana State. He has published a research paper in reputed international journal. He has 3years of experience.

Mr. T. Jagadish kumar, Completed Bachelor of Technology from JNTU Hyderabad University in the year 2011 from Computer Science and Engineering, did his M.Tech-CSE in 2014 from JNTUH. Presently working as Assistant Professor in CSE Dept of Malla Reddy Group of Institutions located in Hyderabad, Telangana State. He has published a research paper in reputed international journal. He has 3years of experience.

Mr. M V Kamal, pursued Bachelor of Engineering from Gulbarga University in the year 2001 from Computer Science of Engineering, did his M.Tech-IT in 2004 and M.Tech-SE from 2010 from JNTUH. Presently working as Associate Professor in CSE Dept of Malla Reddy Group of Institutions located in Hyderabad, Telangana State. He is currently pursuing his Ph.D in CSE from JNTUH. He has published more than 12 research papers in reputed international journals and conference. He has 16+years of experience. He is a member of National and International professional body like computer society of India etc.

