Conceptual Trusted Incident Reaction Architecture

Christophe Feltus

Public Research Centre Henri Tudor
Luxembourg – Kirchberg, Luxembourg
christophe.feltus@tudor.lu

Abstract: Enterprise networks are continuously growing up and rising connections with various software and systems. Their components' security is a tremendous challenge especially due to their heterogeneity and distributed structure. Mechanisms, such as the intrusion detection system, are developed to monitor the security level of those components, their exposure to external attacks or internal failure, and their compliance to target trust level. Although the concept of trust exists for a long time in the computer sciences, it is mainly deployed in the arena of peer-to-peer networking and in specific domains like the eCommerce. The paper proposes a conceptual trusted incident reaction architecture elaborated firstly based on a multi agent system that offers the ability to be dynamic and flexible, and secondly based on a decision mechanism that supports the choice of components based on contextual attributes and based on information weighted with trust value.

1 Introduction

For many years, corporate networks have driven the trends of openness, mobility and flexibility. That anabasis in the field of computer network is an open door for many technical progressions that have rendered possible a new way of making business never imagined before. While that improvement provides many facilities, it also appears to be Pandora's Box for new risks of malicious acts or manipulation problems. The control of a network, its extensions and its progression outside the company is made difficult by these arising new services. Therefore, the supervision and control of the information flow exchange to, and from, significant business functions that it supports are raising a continuously growing amount of sophisticated solutions. Intrusion Detection System (IDS) mostly contribute to expand that set of products at the origin of security reactions. In this paper, we propose a conceptual trust architecture that completes in the first place the traditional itemised requirements. E.g.: react quickly and efficiently to any simple attacks but also to any complex and distributed ones; ensure homogeneous and smart communication between the composing nodes, and be open to a wide range of technology. In the second place, this architecture completes our previous responds to the following additional requirements dictated by new business constraints [GKF09]: ability to make decision on a business based approach, ability to map the solution onto a layered based infrastructure, and integration of the concept of trust in the decision processes. The paper completes our previous works [IGAB06] by including trust in the decision mechanism. This architecture has been defined in a 3 phases approach. The first phase defines

the architecture [GKF09] using a MAS structure modelled on the XACML architecture. The second phase elaborates the decision mechanism supporting that architecture. The decision mechanism is elaborated [FKA10] based on the Bayesian network (BN) and the influence diagram (ID) [Ya07]. The bigger rectangle is the MAS architecture that includes the decision mechanism (shorter rectangle). That last makes decision based on contextual constraints and trust value as input and provides utility value to the node of the system for output.

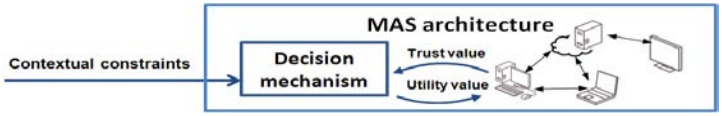

Figure 1: Reaction mechanism architecture

Whereas the here above described architecture permits to react when an incident occurs, it his necessary in parallel to select a language to support the semantic expression of the reaction. Rules elicited by this language compose the system security policy as well as the reaction policy. In case of an incident reaction, policy adaptation is considered as a regulation process. The main steps of the policy regulation process take the business rules as input, and map them onto technical policies. These technical policies are deployed and instantiated on the infrastructure in order to have an improved state of temporary network security stability adapted to the ongoing attack. This policy regulation is thereafter achieved by modifying/adding new policy rules to reach a new set of stable policies.

To illustrate the performance of this reaction architecture, we use the results of the BARWAN project [BKAB98]. 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 [TPDB98] pooled computing power to permit a small number of workstations to support a large number of end users. Fig.2 highlights the distribution of the application over the buildings, the campus and the metropolitan layers. In that paper, an architecture to adapt a reaction once an attack occurs on one of those layers is proposed. Additionally, the architecture makes it possible to integrate internal or external contextual information for the reaction decision like, e.g. the usage of the application, as proposed in the case study, during a medical rescue operation after a serious car accident on the Golden Gate Bridge. The values used for the illustration are issued from [FKA10].

2 Multi agent System architecture

The distributed architecture introduced in the paper is composed of several components, called *operators*, which have different responsibilities. Those operators are organized in two dimensions, as presented in Fig. 2. The vertical dimension, structured in layers relative to the managed network organization, allows adding abstraction in going upward: the

lowest layer is closed to the managed system and thus plays the role of an interface between the targeted network and the management system. The higher layer encompasses a global perception of the whole system and is able to take some decisions based on a more complete knowledge of the system, business, and organization. Intermediate levels (1 to n-1) guarantee flexibility and scalability to the architecture in order to consider management constraints of the 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.

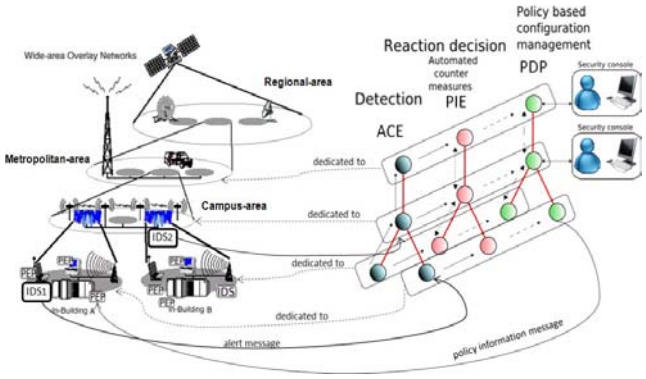

Figure 2: Overall architecture layers

The horizontal dimension contains three trees (alert, reaction and deployment) placed side by side and composed respectively with the following components: (1) The Alert Correlation Engine (ACE) that collects, normalizes, correlates, analyzes the alerts coming from the networks and which represent an incident. The confirmed alert is forwarded to the reaction decision component. (2) The Police Instantiation Engine (PIE) receives the confirmed alert to which a reaction is expected. Considering the knowledge of the policy and of the systems' organization and specified behaviour, these components decide if a reaction is needed and they define that reaction. The reaction may be a modification, an addition to or a removal of current policy. (3) The Policy Deployment Point (PDP) instantiates and deploys the new policies on the targeted networks. The deployment is made by the Policy Enforcement Point (PEP) that enforces these new policies and lead to a new security stability of the network. The terminology used is extracted from both: [Xa00] and [CM03]. Fig. 2. explains how the three layers are mapped to the architecture borrowed from [BKAB98]. From top to bottom: the metropolitan area, the campus area, and the in building network (building A and B).

The scientific literature is rich in papers addressing the usage of XACML as a language to improve the trust in information sharing [MCCP06], however, it lacks proposals concerning the trust between the components of the XACML architecture itself. Despite the few existing solutions, [KCEV05] proposes an architecture based on the addition of a Security Information Point (SIP) along the existing components to manage the following security features: generation of random secret keys for encryption, verification of the integrity of access control policies, management of security related information and

coordination between components. As highlighted in Fig. 3, we complete the SIP with the management of the trust between all of the XACML components. The MAS architecture is associated with a communication engine. This engine is based on a message format and on a message exchange protocol issued by [De07]. The message format is defined in XML and structured around a number of attributes that specify the message source, the message destination, the level on which the destination trusts the source, 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 of rules that governs the syntax, semantics, and synchronization communication. Electronic institution based on agents provides the requisite characteristics to support the function of the operators. Hence, agents are assigned roles in order to specify their function in the architecture and the communication protocol is accordingly defined between them. Fig. 3 introduces the developed architecture illustrated based on BARWAN. The flow is supposed to begin with an alert detected by the IDS, positioned on the InfoPad server. This alert is send to the BuildingA_ACE agent that does or does not confirm the alert to the PIE. The 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. 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 (InfoPad server, fileserver, etc.). Then the PDP agent sends the new policy to the InfoPad PEP agent that knows how to transform a policy into an understandable rule or script for the InfoPad server. The decisions lay on information issued by the context and weighted based on trust values provided by the trust engine. A focused analysis of the PDP points out that it is composed of several modules. For the MAS perspective, the Component Configuration Mapper results from the interaction between the PDP agent and the Facilitator Agent while the Policy Analysis module is achieved 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 (router, firewall, file server, etc.). Therefore, the Facilitator uses *white pages* and *yellow pages* services. The JADE [BPR99] platform provides implemented facilitator and searching services. Besides, the use of a MAS 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. The main goal of the reaction policy enforcement engine is to apply policies in terms of specific concrete rules on “technical” devices (router, firewall, fileserver, and other systems named PEP).

3 Decision Support System

The system should be able to provide mechanisms to make decisions in a set of situations like: conflicts between several choices of reactions or necessities to escalade (or not) reactions to the upper layer. One 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*. The decision inputs of

the alert sending are e.g.: the frequency of the alerts, the contribution of the system to the medical rescue operation (if any), or the criticality of the rescue operation. The decision outputs e.g.: the escalation of the alert to upper ACE. As explained by [Ya07], the decision mechanism stands on four pillars: Ontology, BN, ID and Virtual Knowledge Community (VKC). In that paper, the VKC is not addressed because the 3 first pillars are sufficient to understand the decision mechanism. The preferred approach to design the decision mechanism is studied from the research performed by Yang's thesis and is adapted for the incident reaction through a MAS architecture. This paper completes Yang's research since our DSS is illustrated by a real architecture for incident reaction.

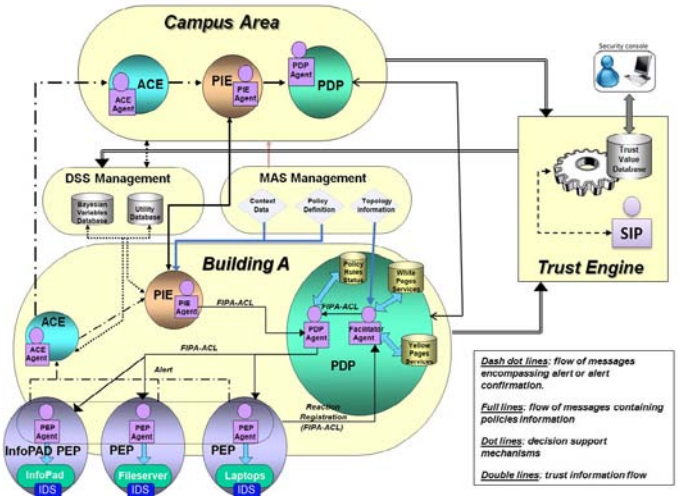

Figure 3: MAS reaction architecture

Ontology is the most import pillar in that it supports the BN and ID pillars. 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 in OWL. It may be implemented in web resources and is structured based on the set [object, subject, trustValue (objet, subject), predicate]. Object and subject are resources, predicate is an attribute or a relation used to describe a resource, and trustValue has been added to reflect the trust value that the subject has on the object. In BARWAN case study, the DSS decides to transfer an alert from the IDS to the BuildingA_ACE, to forward that alert to an upper ACE, and to confirm the alert to the PIE. On Fig 4., $t \in [0,1]$ reflects the trust between the agents that play the role of the components or between these agents and data sources. Fig. 4's data are random for illustration. The ontology permits to formalize the concept encompassed in the MAS architecture as well as their relations. However, at the ontological level of formalization, uncertainty challenge remains unaddressed and decision mechanism remain needed for the agents to take the decision. OntoBayes is an extension of OWL with two features: BN that address the uncertainty and ID that support the decision mechanism process.

ProbCell	*HasPParameters*	*HasPValue*
Cell_1	*alert.severity=low|rescue.impact=low*	*0.8*
Cell_2	*alert.severity=medium|rescue.impact=low*	*0.4*
Cell_3	*alert.severity=high|rescue.impact=l*	*0.1*
Cell_4	*alert.severity=low|rescue.impact=medium*	*0.3*
Cell_5	*alert.severity=medium|rescue.impact=medium*	*0.9*
Cell_6	*alert.severity=high|rescue.impact=medium*	*0.5*
Cell_7	*alert.severity=low |rescue.impact=h*	*0.1*
Cell_8	*alert.severity=medium|rescue.impact=high*	*0.4*
Cell_9	*alert.severity=high|rescue.impact=high*	*0.7*

Table 1: Bayesian probability

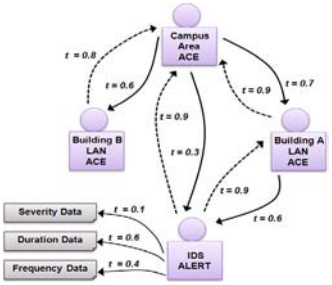

Figure 4: Inter components trust values

The Bayes theorem is used to calculate conditional probabilities. The calculation depends on prior knowledge that could be considered as uncertain. E.g.: the probability of high impact on the medical rescue if we have a medium severity alert beforehand. The BNs extension introduces the parameters of that probability by specifying the following two perspectives: qualitative and quantitative. The qualitative perspective specifies the random variables explicitly as well as their dependencies and the later links quantitative information to those variable**s** using OWL. The specification of random variables and their dependency is performed by introducing the new OWL property element `<owl:ObjectProperty rdf.ID="dependsOn"/>` [Ya07]. Accordingly, the qualitative extension may be represented by 2 Bayesian graph models (Fig. 5). The ovals represent Bayesian variables and the arrows specify their relations. The graph is to be read, e.g. 1.: The alert that is forwarded from the BuildingB_ACE to the network upper ACE has influence on the confirmation of the alert that is send from the CampusArea_ACE to the CampusArea_PIE. E.g. 2.: The severity of the alert has influence on the action to send an alert to the BuildingA_ACE. The last examples may be translated using the new OWL `dependsOn` element as follows: The quantitative extension is performed in association with the probability table of the Bayesian variables. In case of the BARWAN, Table 1 provides the quantitative probability P (Table I).

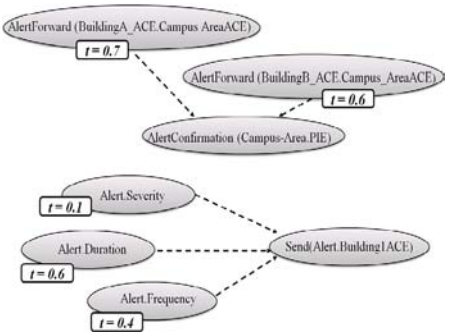

Figure 5: Bayesian graph models

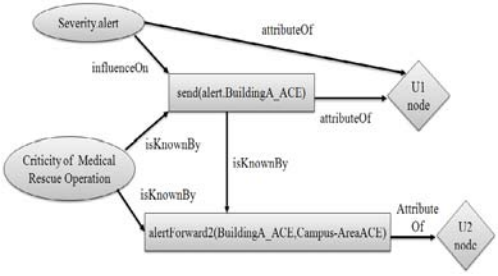

Figure 6: ID's graph model of alert transfer

Utilitv	*HasUParameters*	*HasUValue*
Cell_1	*send(alert.BuildingA_ACE)=yes|severity.alert=low*	*-80*
Cell_2	*send(alert.BuildingA_ACE)=yes|severity.alert=medium*	*50*
Cell_3	*send(alert.BuildingA_ACE)=yes|severity.alert=high*	*100*
Cell_4	*send(alert.BuildingA_ACE)=no|severity.alert=low*	*80*
Cell_5	*send(alert.BuildingA_ACE)=no|severity.alert=medium*	*40*
Cell_6	*send(alert.BuildingA_ACE)=no|severity.alert=high*	*-100*

Table 2: Utility for in lan ACE alert sending

IDs extension aims at representing and analyzing a decisional model to support the decision making process. The review of the literature dealing with the issue of ID [HM05, Je01] underlines that decision mechanisms are composed by three types of nodes: 1) *Chance nodes* that represent variables, which are not controlled by the decision maker,

2) *Decision nodes* that represent choices available for the decision maker, and 3) *Utility nodes* that represent agent utility functions. Additionally, [TS90] explains that three types of arcs express the relationship between the nodes: I) *Information arcs* (`isKnownBy`) that point out the information that is indispensable for the decision maker, II) *Conditional arcs* (`influenceOn`) that point out the probabilistic dependency on the associated variable, and III) *Functional arcs* (`attributeOf`) that point out variables used by utility nodes as decision criteria. Based on that structure of a decisional model, the alert transfer may be represented in Fig. 6. Ovals stand for *Chance nodes*, rectangles stand for *Decision nodes*, and diamonds stand for *Utility nodes*. The information arc relates to all information observed to make a decision and the conditional arc relates to data issued from Chance node and considered as evidence for the Decision nodes. Additionally, to make a decision, the agent that takes the decision needs to have its preferences quantified according to a set of attributes. The most important preference has the higher value whereas the worst has the lower one. To achieve that, the Utility node is associated with a utility table that gathers the preferences for all decision choices. Table 2 shows these preferences for the BuildingA_ACE alert sending decision and is represented by the utility database in Fig. 3. As seen in Fig. 6, a sequential path between all decisions exists. Indeed, some decision depends on previous decisions and as a consequence, previous decisions (decision node) become chance nodes for next chance node. This figure illustrates that send(alert.BuildingA_ACE) is at the same time a decision node and a chance node that is known to be the decision node alertForward2(BuildingA_ACE,CampusAreaACE).

The analyze of the DSS shows that, according to the BARWAN case study, the probability of having a high impact on the rescue is meaningful (0.7) if the severity of the alert is high (Table I, Cell_9). Hence, sending the alert to the BuildingA_ACE when the alert severity is high has much utility (Table II, Cell_3). However, in the decision process, the trust parameter is also to be taken into account: t=0.1 (for the severity parameter). That reflects, E.g. that the severity is often badly evaluated. For others parameters like the alert duration, the trust level is higher (t=0.6). As a consequence, if we suppose that the impact on the rescue of the alert duration is important, this parameter will be more meaningful, to the decision to send the alert to the BuildingA_ACE than to the severity. On the contrary, if its impact on the rescue is low, its value will accordingly be reduced. The impact of trust in the above paragraph is based on two different parameters (severity and duration). Trust is moreover significant for a decision based on parameters that provides the same information. E.g., in Fig. 6, the alert is confirmed by the CampusArea_ACE to the CampusArea_PIE based on the alert forwarded from the BuildingA_ACE (t=0.7) and from the BuildingB_ACE (t=0.6). In that case, depending on the configuration of the DSS engine, the decision may be taken whether based on the more trusted agent or on the average values weighted by trust. This possible to use trust in IDS offers the advantage to refine the decision make by the MAS not only regarding the value send by an agent but also based on the context in which the agent evaluates or based on the previous data he has provided.

Although MAS has already been largely investigated in the field of crisis management, linking the decision making process with trust values remains to the best of our knowledge not significantly addressed. The review of the research performed in that field

reveals the plethora of efforts made to enhance the detection of attacks and to correlate them with vulnerability databases [La99], to automate the reaction [BP04], and to improve their performance [Ja99,AAPM04]. However, research aiming to ensure a global reaction to attacks in order to avoid their propagation and/or to help the administrator to deploy the appropriate reactions, remains restrained to some very specific applications and domains like [RJCM03] that focuses on web services and internet servers or [TCGN06] that proposes a protocol named *ContagAlert,* which is able to propagate an alert while an attack is in progress. This protocol uses contagion spreading behaviour and is consequently well tailored for wide spread network. The inconvenient of it is that the decision for the alert propagation is based on threshold behaviour and do not integrate the business constraints in the decision mechanism. [IAA06] establishes a connection between the business and the technology but the perspective of its analysis rather concerns the value associated with a well thought IDS deployment strategy than IDS systems tailoring according to the business services. [WFCR01] also proposes a cost benefit analysis for IDS that reflects the business needs but does not accordingly parameterized the IDS. Trust and MAS in IDS has been recently introduced by [BJGD04] and [RTPP07] under a technical perspective, but in those last researches, as well as in the previously depicted, the alignment of a solution utility with the business value has been omitted.

4 Conclusions

The paper presents a conceptual trusted incident reaction architecture based on a policy regulation approach strategy. The solution is composed firstly with a MAS that offers the advantage to react quickly and efficiently to an attack while being adapted for heterogeneous and distributed networks. Secondly, with a decision support system that helps agents to make decisions based on utility preference values and new requirements coming along those architectures: the awareness of contextual information and the integration trust weighted attributes. The architecture has been illustrated based on BARWAN. Accordingly, the decision mechanism has been analyzed for the criticality of the medical rescue operations while taking the trust in the IDS component into account. Future works focuses on analysing the performance of the architecture.

This research was funded by the National Research Fund of Luxemburg in the context of TITAN (Trust Assurance for Critical Infrastructures in Multi agents Environments, FNR CO/08/IS/21) project.

References

[AAPM04] S. Antonatos, K.G. Anagnostakis, M. Polychronakis, E.P. Markatos: Performance analysis of content matching intrusion detection systems. 4th IEEE/IPSJ 2004.

[BJGD04] J. Bigham, X. Jin, D. Gamez, I. Djordjevic, C. Phillips: Dynamic Trust Management of Semi-Automated Complex Systems. CCCT'04, Austin, USA, August 2004.

[BP04] Z.K. Baker; V.K. Prasanna: Automatic synthesis of efficient intrusion detection systems on FPGAs, 14th Int. Conf. Field Program. Logic Appl., 2004, pp. 311-321.

[BPR99] F. Bellifemine; A. Poggi; G. Rimassa: JADE - A FIPA-compliant agent framework, CSELT internal technical report. PAAM'99, London, April 1999, pp.97-108.

[BKAB98] E.A. Brewer, R.H. Katz, E. Amir, H. Balakrishnan, Y. Chawathe, A. Fox, S.D. Gribble, T. Hodes, G. Nguyen, V.N. Padmanabhan, M. Stemm, S. Seshan, T. Henderson: A network Architecture for Heterogeneous Mobile Computing, IEEE Personal Communications Magazine.

[CM03] F. Cuppens, A. Miège: Modelling contexts in the Or-BAC model, 19th Annual Computer Security Applications Conference, Las Vegas, December, 2003.

[De07] IDMEF/RFC4765, Network Working Group: Debar, H., France Telecom; D. Curry, Guardian; B. Feinstein, SecureWorks, Inc.; March 2007.

[FKA10] C. Feltus, D. Khadraoui, J. Aubert: A Security Decision-Reaction Architecture for Heterogeneous Distributed Network, IEEE ARES 2010, Krakow, Poland.

[GKF09] B. Gâteau, D. Khadraoui, C. Feltus: Multi agents System Service based Platform in Telecommunication Security Incident Reaction, IEEE GIIS 2009.

[HM05] R.A. Howard, J.E. Matheson: Influence diagrams. Decision Analysis, 2(3):127–143.

[IAA06] C. Iheagwara, F. Awan, Y. Acar, C. Miller: Maximizing the Benefits of Intrusion Prevention Systems: Effective Deployments Strategies, 18th FIRST 2006, Baltimore.

[IGAB06] C. Incoul, B. Gateau, J. Aubert, N. Bounoughaz, C. Feltus: If only I can trust my police! SIM : an agent-based audit solution of access right deployment through open network, CRISIS 2008, Tozeur, Tunisia.

[Ja99] K.A. Jackson: Intrusion Detection System (IDS) Product Survey, Distributed Knowledge Systems Team; Information and Communications Division; 1999, Los Alamos National Laboratory, Los Alamos, NM.

[Je01] F.V. Jensen: Bayesian networks and decision graphs. Springer, corr. print. ed, 2001.

[KCEV05] Y. Keleta, M. Coetzee, J.H.P. Eloff, H.S. Venter: Proposing a Secure XACML architecture ensuring privacy and trust, 5th ISSA, July 2005, ISBN 1-86854-625X.

[La56] H.D. Lasswell: The decision process; seven categories of functional analysis, College of Business and Public Administration, University of Maryland, 1956.

[La99] L.J. LaPadula: State of the Art in Anomaly Detection and Reaction Technical Report MP 99B0000020, Mitre, July 1999.

[MCCP06] U.M. Mbanaso, G.S. Cooper, D.W. Chadwick, S. Proctor: *Privacy Preserving Trust Authorization Framework Using XACML.* UNSPECIFIED, ed. International Workshop on Wireless Mobile Multimedia. IEEE Computer Society, pp. 673-678.

[RJCM03] J.C. Reynolds, J. Just, L. Clough, R. Maglich: On-Line Intrusion Detection and Attack Prevention Using Diversity, Generate-and-Test, and Generalization, HICSS'03, p.335.2.

[RTPP07] M. Rehak, J. Tozicka, M. Pechoucek, M. Prokopova, L. Foltyn: Autonomous Protection Mechanism for Joint Networks in Coalition Operations, KIMAS 2007.

[TS90] J.A. Tatman, R.D. Shachter: Dynamic programming and influence diagrams. IEEE Transactions on Systems, Man, and Cybernetics, 20(2):365–379, 1990.

[TCGN06] M. Treaster, W. Conner, I. Gupta, K. Nahrstedt: ContagAlert: Using Contagion Theory for Adaptive, Distributed Alert Propagation, 5th IEEE NCA 06.

[TPDB98] T.E. Truman, T. Pering, R. Doering, R.W. Brodersen: The InfoPad Multimedia Terminal: A Portable Device for Wireless Information Access. IEEE Trans. Comput. 47, 10, 1998.

[WFCR01] H. Wei, D. Frinke, O. Carter, C. Ritter: Cost-benefit analysis for network intrusion detection systems, 28th CSI, Washington, USA, 2001.

[Ya07] Y. Yang, A framework for decision support systems adapted to uncertain knowledge, PhD. Thesis, 2007. University of Karlsruhe.