Adaptive Security and Trust Management for Autonomic Message-Oriented Middleware

Habtamu Abie
Norwegian Computing Center
Department of Applied Research in Information Technology (DART)
Oslo, Norway
Habtamu.Abie@nr.no

Abstract—**With the increase in society's dependence on IT communication systems, the need for dependable, trustable, robust and secure adaptive systems becomes ever more acute. Modern autonomic message-oriented middleware platforms have stringent requirements for self-healing, adapting, evolving, fault-tolerance, security and active vulnerability assessment, especially when the internal working model of a system and the environmental influences on the system are uncertain during run-time. In this paper we present an adaptive and evolving security (AES), and an adaptive trust management (ATM) approach to such autonomic messaging middleware systems, an approach that learns, anticipates, evolves and adapts to a changing environment at run-time in the face of changing threats. The methodology used in this research is partly analytical and partly experimental. This involves analysis of how the principles of AES and ATM can be applied to the environment resulting in the development of theoretical models which are then tested in practice by prototyping.**

Keywords-Adaptive security; trust management; self-healing; resilience; evolution; security monitoring;security metrics

I. INTRODUCTION

The environment surrounding modern communication and information systems is in a continuous state of change throughout the lifetime of an application. Autonomous adaptive systems deal with the uncertainty, which is ascribable to a number of different factors, by being self-organizing and emergent. Self-organization is achieved when the system constructs, and adaptively maintains, its own behavior without external control. Emergence is the state of a system when it exhibits coherent system-wide or macroscopic behavior that is generated dynamically by the local interactions between the individual entities at the microscopic level. Note that the term macroscopic refers to the dynamics of the system as a whole, while the term microscopic refers to the dynamics and doings of the individual entities within the system.

Message-oriented middleware (MOM) enables applications to exchange messages with other applications, without having to know details of the other applications' platforms and networking, thus increasing the interoperability, portability, and flexibility of architectures. Because they must be self-organizing and emergent, modern autonomous MOM platforms have stringent requirements for self-healing, adapting, evolving, fault-tolerance, security, and active vulnerability assessment, especially when the internal working model of a system and the environmental influences on the system are uncertain during run-time.

GEMOM (Genetic Messaging-Oriented Secure Middleware) [1] addresses these issues and provides solutions to overcome limitations in robustness, resilience, evolvability, adaptability, scalability, and assurance against vulnerabilities to security threats and erroneous input during run-time in the face of changing threats. In this study we have developed adaptive and evolving security (AES), and adaptive trust management (ATM) models which are essential for such an autonomic MOM system, models that learn, anticipate, evolve and adapt to the changing environment at run-time in the face of changing threats without sacrificing too much of the efficiency, flexibility, reliability and security of the system.

In this paper we present briefly the GEMOM-enhanced secure, resilient and reliable MOM, and our AES and ATM models, before discussing prototyping and related work, and presenting our conclusion and future perspectives.

II. SECURE, RESILIENT, AND RELIABLE MOM

GEMOM abstracts out the notion of "fault", and fault tolerance is looked at in a more dynamic way. GEMOM's definition of intelligence and resilience draws attention to the fact that there can be insensitivity to or low awareness of faults. These faults can result in the deterioration of the functional profile of the information system, of the volumetric profile, or of the security profile. It also brings up the question of the availability of support for a reconfiguration back to an efficiently working system. GEMOM is able to rectify such vulnerability to faults by researching, developing and deploying a prototype of a messaging platform that is evolutionary, self-organizing, self-healing, scalable and secure. GEMOM is resilient and utilizes redundant modules (hot-swap or switch-over) instantly without information loss. These resilience features allow specialist, independent system actors, (viz. watchdogs, security and situation monitors, routers, and other optimizers) to remove or replace compromised nodes in the broader network instantly and without compromising higher levels functionality and security.

978-1-4244-5113-5/09/$25.00 ©2009 IEEE

Existing technologies are crude, not scalable and not suited to future needs. They have neither the robustness nor the resilience appropriate for future real-time systems in particular, and GEMOM provides solutions to overcome these limitations to secure autonomic messaging [1]. GEMOM has made and is making advances in the following areas: resilience, self-healing, scalability, integrated vulnerability management, better interoperability and integration of distributed systems, and holistic and systematic adaptive security monitoring and measurement.

A. *Resilience, Self-healing and Scalability*

Resilience and self-healing are achieved by the use of an overlay of brokers that supports resilience in systems that depend on publish/subscribe MOMs, despite the lack of any privileged knowledge of the underlying infrastructure. The brokers in the overlay are called GBrokers (G-Nodes). Fig. 1 depicts such a G-Node. A Broker Overlay Manager Agent has been developed that performs autonomous adjustments to the run-time configuration of the system in order to preserve and maintain optimal and uninterrupted operation, also in case of partial breakdowns. It supports mechanisms for

- adding G-Nodes, for measuring QoS between overlay components and publishers and subscribers and deciding what action to be taken to mitigate loss of QoS or breakdowns;
- discovering and communicating with other components in the overlay network;
- evaluating the performance of the system in the context of the monitored performance;
- establishing the state of the overlay network; and
- Making decisions on the reconfiguration of routing and message passing.

It also learns from experience and uses its new knowledge in its prediction and decision-making. Two approaches are used to achieve resilience and evolution, one being the management of reserve resources in such an overlay network, the other being empirical correlations.

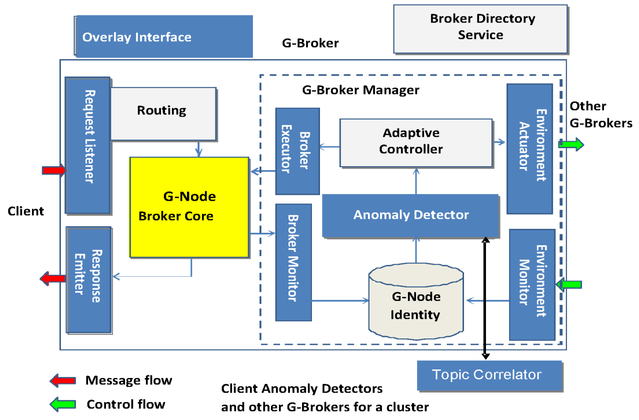

Figure 1. Adaptive overlay of G-Node

In GEMOM, scalability and resilience are achieved via cooperating brokers, publishers and subscribers with sufficient replication of paths and namespaces, and clustering of topics into groups of one or more with group replication. This allows the system to avoid overloading specific brokers, and to sustain random and sudden fallout without any interruption of service.

B. *Vulnerability Management*

The innovation represented by the GEMOM vulnerability management system is the integration of the detection systems, intelligent techniques, and the threat and vulnerability management tool-set into the management system. The detection systems include mechanisms for the detection of security vulnerabilities, input errors, misconfiguration errors, and bugs. The intelligent techniques include the search and discovery of vulnerabilities and other errors, and the detection of violations of QoS and privacy policy. The threat and vulnerability management tool-set provides mechanisms for threat discovery, and techniques to support generic, intelligent, adaptive approaches to robustness and security testing in a distributed environment.

Knowledge of the different kinds of vulnerabilities, the software functions, aspects of the semantics of the application domain, and protocols used, is integrated into the tools in order to find vulnerabilities and errors in the software [1].

C. *Enhanced Interoperability and Integration*

The Publish/Subscribe variant of MOM is an efficient mechanism to integrate distributed systems. This messaging paradigm provides key properties for efficient system modeling, viz modeling and re-factoring of the system during run time, and making the system inherently extensible. This paradigm is also a powerful base for implementation of scalability and resilience. In addition, GEMOM supports better interoperability and integration of information systems by allowing actual instances to be configured so that various functions are subcontracted to one or more separated, external or federated entities. This separation allows for a different security layout of different individual, or clusters of, services. For the advantages of this approach, see [1].

D. *Holistic and Systematic Adaptive Security*

Existing MOM systems are not able to guarantee holistic and systematic security, privacy and trust management [1]. As GEMOM advances in the areas described above, an adaptive, holistic and systematic security approach is necessary to meet GEMOM's stringent requirements for self-healing, adapting, evolving, fault-tolerance, security, and active vulnerability assessment. Thus the GEMOM security solution consists of a continuous cycle of monitoring, measurement, assessment, adaptation and evolution to meet the challenges in the changing environments. The main components are described in Chapters 3 and 4.

E. *Autonomic and Genetic Makeup*

Some of the G-Nodes are operational nodes and some managerial. Fig. 2 shows layers of such G-Nodes. The operational nodes can be classified as producer/publisher, consumer/subscriber and broker nodes with their specialized agents such as sensors, effectors, monitors, detectors, analyzers, etc. The sensors and effectors communicate with managerial nodes of different types. Managerial nodes can be classified as QoS Mangers, Resilient Managers, Security Anomaly Managers (the Anomaly Detectors would be in the operational nodes, but there can be many and they can be layered), Adaptive Security Managers, etc. The managerial nodes make decisions about the run time operation of the system that require a wider perspective than the individual operational nodes have. Each node is aware of its context, dynamically adapting itself to continuously evolving situations, and maintains integrity by reacting to known changes, adapting to unknown changes, or dying.

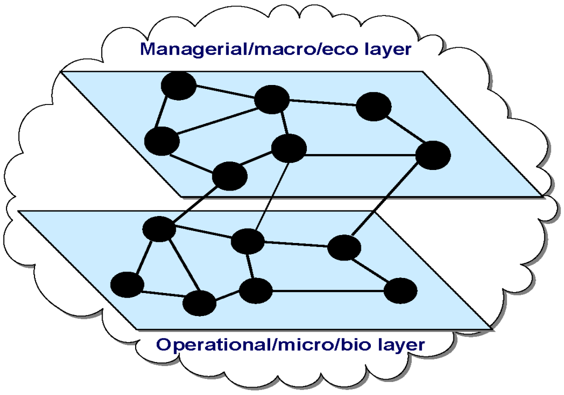

Figure 2. Layers of G-Nodes

The biological and ecosystem metaphors provide interesting parallels to the conceptualizations and descriptions of the G-Nodes. The overall GEMOM system architecture has a structure similar to that of a complex adaptive system that utilizes autonomic systems that mimic biological auto-immune systems at the microscopic level (operational level in this case) and that utilize the behaviors of an ecosystem of disparate entities at the macroscopic level (managerial level in this case). Biological and ecological systems maintain system integrity by reacting to known changes, adapting to unknown changes, or dying. The adaptations and responses can be at a macroscopic ecosystem level (e.g., system or species) or a microscopic biological level (e.g., molecular, cellular) [22]. Hence we can consider GEMOM as having a genetic makeup [1].

III. Adaptive and Evolving Security

Modern autonomous MOM platforms have stringent requirements for adaptive and evolving security, especially when the internal working model of a system and the environmental influences on the system are uncertain during run-time. Adaptive security refers to a security solution that learns and adapts to the changing environment at run-time in the face of changing threats, and anticipates threats before they are manifested. Evolving security refers to the modification of existing security functions and the generation of new security functions for long-term adaptivity in a non-disruptive way.

Consequently GEMOM has developed an AES approach to meet these requirements, and to maintain the proper balance between security and performance in rapidly changing environments. Such an approach involves gathering contextual information both from within the system and from the environment, measuring security level and metrics, analyzing the collected information and responding to changes (a) by adjusting internal working parameters such as encryption schemes, security protocols, security policies, security algorithms, different authentication and authorization mechanisms, changing the QoS available to applications, and automating reconfiguration of the protection mechanisms, and/or (b) by making dynamic changes in the structure of the security system [19], [4]. The analysis part of such an approach requires flexible learning and decision making processes for parametrical, structural and goal adaptation that help set priorities and make the best decision when both the qualitative and the quantitative aspects of a decision need to be considered.

The AES security services must adapt to the rapidly changing contexts of the GEMOM environment. The AES model consists of a continuous cycle of monitoring, assessment, and evolution to meet the challenges in the changing, multifaceted relationships within and between organizations both in autonomic MOM-based business environments and today's rising threat situation. The AES model utilizes contextual information and decision making to select the "best" security model for a given situation. The AES includes the integration of monitoring, analysis and response functions, and tool-set, elastic/fine-grained adaptive authorization, adaptive authentication and Federated Identity Management, and tools and processes for pre-emptive vulnerability testing and updating.

A. Theory of Security Adaptation

We used [19] as the basis for the theory of our security adaptation model, a schematic representation of which is shown in Fig. 3. Adaptation may be defined as the optimal control (i) of a specified object F in a state S whose influence Y on the environment is determined by the influences X of the environment on the object, (ii) of the relevant set of adaptable structures and/or factors U, and (iii) of the goals Z of the adaptation as defined by specified constraints on the state S of the object. The mathematical formalization and the adaptive algorithms that can learn and change their behavior by comparing the results of their actions with the goals that they are designed to achieve, are defined in [19].

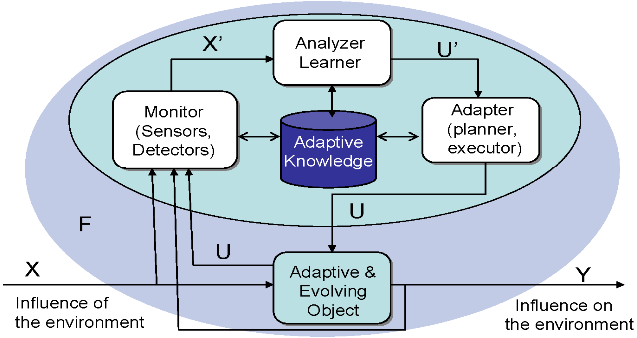

Figure 3. Adaptive and evolving security model [19]

B. *Adaptive Security Manager*

The core component of the AES model is the Adaptive Security Manager (ASM), which manages and controls all the security components as an integrated GEMOM security infrastructure. Fig. 4 shows the main components of the ASM model. All these main components, with the exception of the Policy Manager and Analyzer, have been prototyped are currently being tested. While each component implements a local adaptation control loop, the ASM implements a global adaptation control loop. Here the Sensors are Anomaly Detector, Security Monitors, Fault Detectors, QoS Monitors, Audit and Logging, etc, and are described in the ensuing sections. Each component owns a public and private key pair by means of which it can sign and encrypt messages, and a certificate to attest their identity. The identity certificate, containing inter alia the component's principal name and the name of the owner, and is signed by the Key Management Framework (KMF) which acts as a Certification Authority (CA) or Source of Authority (SOA) to guarantee the authenticity of the certificate.

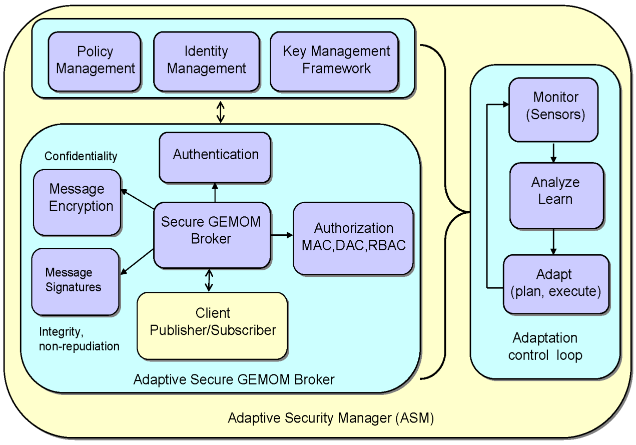

Figure 4. GEMOM AES model

C. *Self-Protection*

A self-protecting system, as defined by IBM [6], can anticipate, detect, identify, and protect itself against threats, unauthorized access, and denial of service attacks. Therefore, GEMOM as an autonomic MOM has to implement self-protecting capabilities that can detect hostile behaviors as they occur and take corrective actions to make itself less vulnerable. In the GEMOM setting, the self-protection is handled either by a single entry point (micro property) which gives each node authorization, by a coordinated defensive group attack by the other nodes (a macro property), or by a combination of the two (defense-in-depth). Most intrusions can be handled by triggering a one-shot behavior of the GEMOM system. However, the GEMOM system has to be constantly alert, so the degree of protection over time (ongoing) is important. Based on the structure, the AES self-protection can be decomposed into three levels (threat points) whose granularity is summarized in Table 1. The three levels work together to achieve the necessary self-protection of the GEMOM system.

TABLE I. GEMOM SELF-PROTECTION

Granularity level (threat point)	**Self-protection**	
	Issues	***Solutions***
Communication/ network level	Protection from malicious node	Network level self-protection mechanisms. Network level trust management scheme. Confidentiality, integrity or authenticity of underlying IP-network can be guaranteed using TLS/SSL connection between routing nodes. Trust models at this level help assess the quality of new joining nodes and the degree of confidence in their behaviours. Anomaly-Based Self-Protection [24].
Broker Nodes level	Protection of run-time environment	Trusted execution environment for nodes. Node self-protection such as mutual authentication and authorization of broker nodes for accurate namespace resolution in order to protect against threats from rogue brokers and to protect confidentiality and integrity.
Publisher/ subscriber level	Protection from malicious publisher/ subscriber	Security contracts or service level agreements. Use of authentication and sub-set of mechanisms to enforce access control for authorized publishers/subscribers. Node level trust management scheme such as certificate- or token-based, and adaptation and maintenance of the trust level over time by building a reputation feedback mechanism.

D. *Adaptive Integration Architecture*

As already stated, the AES includes adaptive integration functions and tool-sets. This section briefly describes the integration of these tool-sets using adaptive authorization as an example of how these tools can be integrated. Fig. 5 shows the adaptive integration architecture. All these tool-sets, with the exception of the last two mentioned under "Adaptive Tools", have been prototyped and tested.

The Adaptive Authorization component provides adaptive authorization through changing security policies, algorithms, protocols and encryption schemes according to context parameters, such as environment, system threats, user threats, trust levels, usage, security and trust metrics, faults, quality of service, etc. Fault and intrusion tolerance mechanisms are used to increase the availability of a system, and previous faults caused by the user are used to increase suspicion-level. The system threat-level and the user suspicion-level are maintained by and obtained from the Adaptive Tools (Security Monitor, Anomaly Detector, Fuzzing Tool, etc). The Adaptive Authorization component allows trust building by allowing the gradual establishment of trust based on attributes, credentials, identities and anomalies, and attack-based trust models.

The Adaptive Analyzer component analyses the collected information using established analysis and decision making methods. It processes the collected data, along with other information (e.g. security policy, threat levels, or trust levels boundaries) and proposes actions to bring about a new stage. The Adaptive Tools sense and gather contextual information both from within the system and from the environment, and distribute information about the security environment to the

Adaptive Analyzer and adaptive database. The Vulnerability Discovery Toolkit allows the identification and understanding of the risks and vulnerabilities of the GEMOM system and the forming of trust solutions to address the risks and vulnerabilities. The Fuzzing Tool allows an effective black box testing technique to be used for finding security flaws from software.

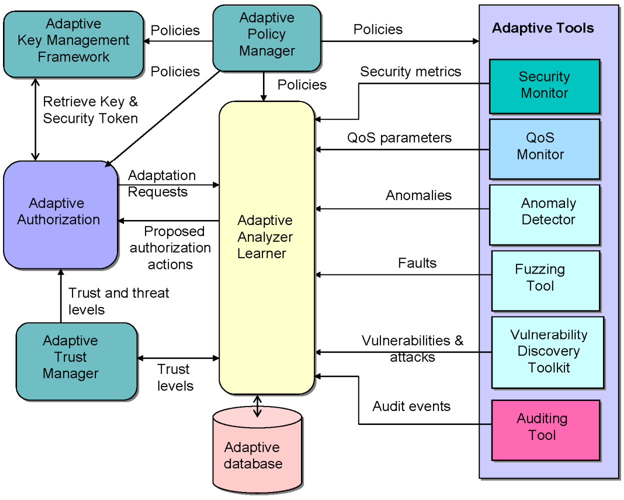

Figure 5. Adaptive integration tools

E. *Security Monitoring and Measurement*

The security and QoS management of GEMOM is based on monitoring [17]. Fig. 6 shows the Monitoring Tool that has been prototyped and tested. The monitoring functionalities are developed and validated against different scenarios of the changing environment. The developed Monitoring Tool supports both the QoS Manager and the Security Manager, including metrics, thresholds and suitable mechanisms for the collection of evidence. The QoS Manager and Security Manager do the actual measuring and make decisions, while the Monitoring Tool carries out tasks such as the management of measured data, threshold, averaging, etc. Some of the sensing in the GEMOM system is done via QoS measurements. The QoS Manager and Resilience Manager resolve problems in their domains by using QoS measurement.

Security measurement is done using both on-line and off-line metrics. The combined use of on-line and off-line metrics supports adaptive measurement: results from applying the off-line metrics are used to re-configure the on-line metrics. The detection of anomalous and the monitoring behavioral patterns (reactive operations), and the use of up-to-date information about threat, vulnerability and reputation levels that are processed using off-line metrics (proactive operations) form the basis of the online metrics. The use of both off-line and online metrics enables us to cope with changing threat levels, and to develop the system further so that it can achieve and assure security over time [16], [17]. The discovery of anomalies, i.e., patterns that are anomalous to constructed (learned) models of normal characteristics, is the task of the Anomaly detector. The learning is the task of the Profiler component. While QoS data are fed to the QoS Manager, the Adaptive Security manager and the Security Measurement Manager, the outputs from the Anomaly Detector are fed to the Security Measurement Manager.

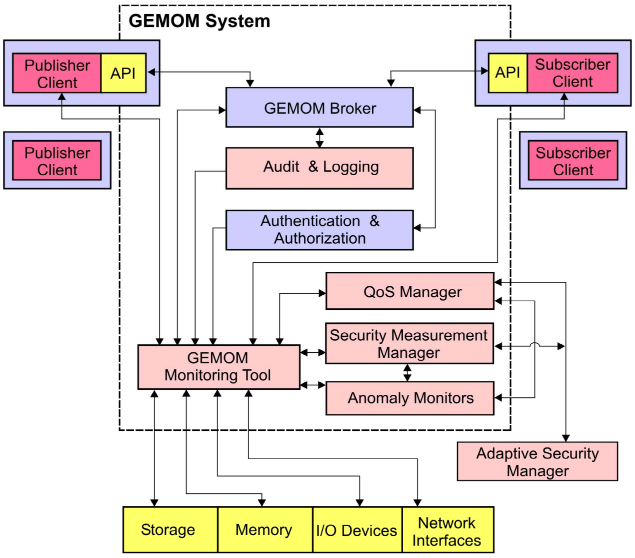

Figure 6. GEMOM monitoring and adaptive security management [17]

IV. Adaptive Trust Management

The GEMOM ATM model is logically organized into a security-based model and a compromised-based model. These two models work together to achieve the adaptability of trust and security in GEMOM. Fig. 7 shows these two models and the interaction between them. Our compromised-based trust model is inspired by the recognition of the crucial role played by the assessment and management of trust, and by the rejection of the assumption that trust relationships are binary and static in nature and that models based on such an assumption are good approximations to real-life computing situations, expressed in [20]. The Security-based trust model is achieved via the security services of the AES, which support the establishment of trust through the provision of a secure and trustworthy environment.

By concentrating on the notion of a “fault”, the GEMOM project expects to make advances in the security of messaging. In addition to understanding intuitively the nature of a fault that stops any actor being operable, leads to a connection being lost, etc, the GEMOM project extends the notion of fault to include compromised security or the unavailability of adequate bandwidth in the first iteration. The final iteration also includes the abstract notion of a “compromised-resource”. Therefore, we developed a compromised-based trust model for GEMOM based on [20]. Our trust model provides information about any attack on the system and the nature of that attack for the purpose of establishing whether, and if so how, different properties of the system have been compromised. In addition, it establishes whether these properties can be trusted for a particular purpose in spite of being compromised and to what degree these judgments should be suspected or monitored. The trust model is organized into three levels and the three levels work together to achieve its adaptability.

The overall ATM system is designed to adapt to the dynamism of the GEMOM environment and to changing degrees of compromise in the GEMOM components by deciding dynamically which approach is to be adopted, and to provide the best likelihood of achieving the greatest benefit for the smallest risk, i.e, maximizing the value of taking a risk.

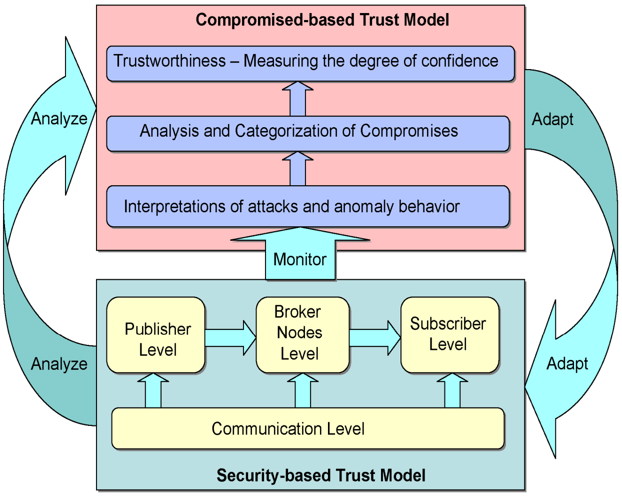
Compromised-based Trust Model
Trustworthiness – Measuring the degree of confidence
Analysis and Categorization of Compromises
Interpretations of attacks and anomaly behavior
Monitor
Analyze
Adapt
Publisher Level
Broker Nodes Level
Subscriber Level
Analyze
Adapt
Communication Level
Security-based Trust Model

Figure 7. GEMOM adaptive trust model

A. Level 1 – Interpretations of Attacks and Anomalous Behaviors

At this first level, the necessary information is collected, filtered and organized for the purpose of triggering analysis and inference. It is not the precise nature of the attacks and anomalous behaviors that have taken place that is of primary interest, but rather their value as an indication of how the system may have been compromised. It is our intention, on the basis of the work done by the Intrusion Detection System (IDS) community and others, to produce annotated taxonomies of different types of attack and to determine how they can be fed into the next level of the trust model [20]. As part of this work, we developed a holistic framework for security metrics development based on threat and vulnerability analysis, security requirements and use case information [17]. The adaptive security tool-sets developed in the project are used for the interpretations of attacks and anomalous behaviors.

B. Level 2 – Analysis and Categorization of Compromises

At this level we address the following three issues:

- Standard security protections (privacy, integrity, authentication, authorization, confidentiality, non-repudiation, etc) and operational properties such as QoS are used to classify compromises. For this purpose, we have developed a framework for the identification of basic measurable components, and a process for the development of security metrics [17].

- Control properties - DoS or other compromises of the system - such as (i) the degree of confidence, (ii) the observability of security and operational properties, and (iii) the degree of controllability (loss of observability) - can come about through attacks on the GEMOM monitoring system. These are observed by the Adaptive tool-sets described above.

- Distinguishing different sub-entities - G-Nodes communication network, the operational processes, the security mechanisms, more finely in order to distinguish the types of information, operations, information sources or destination affected, and the identity and roles of entities participating in, or affected by, the compromises.

C. Level 3 – Trustworthiness

At level we measure the degree of confidence in the security mechanisms used, in the interpretations of attacks and anomalies, in the analysis and categorization of possible compromises and their possible impact, in our ability to make rational decisions, in the adaptation achieved, and in the maximization of the value of taking risk. For this, we collect knowledge of (a) attack types, to guide our attempts to defend against future attacks, (b) compromises, to indicate the threats to operations, and (c) trust, to state guides as to how GEMOM carries on in the face of partially understood compromises.

For the measuring and calculating of the degree of confidence, we have proposed, discussed and developed a flexible framework for the assessment and calculation of the degree of the trustworthiness of and confidence in the measurement of the overall security level of the system as a whole [16]. The framework is dynamic and adaptive, depending on the behavior and security measurement results of the measurable components.

We assess the trustworthiness of and confidence in the overall GEMOM system according to this trust model in order to make decisions about how to adapt to rapidly changing environments. The trust model gives a probabilistic representation of the trustworthiness of and confidence in each measurable component in the system.

To ensure that the decisions we make represent a good basis for secure adaptation, we are developing theoretical reasoning and decision making models by exploiting the synergy effect of the Bayesian inference networks, Genetic algorithms, Analytic Hierarchy Process (AHP), Multi-attribute Decision making (Dempster-Shafer - D-S), Case-based reasoning, and other established learning engines [8].

V. PROTOTYPING AND VALIDATION OF RESULTS

Since prototyping is a tried and true method of capturing the details of the design of a system, in this research we used prototyping to explore design alternatives, test theories and confirm performance. We used our experience to tailor the prototype to our specific requirements. Our prototypes are being used both to confirm and verify user requirements that our design must satisfy through case studies, and to verify the performance and suitability of our design approach. Following the common strategy of the GEMOM project – i.e., design, test, evaluate and then modify the design based on the analysis of the prototype, we have developed, prototyped and lab-tested a full-featured message broker, transparent completion and encapsulation publishing framework, adaptive security implementation (authentication, authorization, key management, Identity management), a MOM Intelligent Fuzzing Tool for a pre-emptive security "black box" testing, a Security Monitoring Tool described in section "III.E" and shown in Fig. 6, and tools for the management of configuration and deployment and development process.

We have also developed demonstrators for enhanced resilience, QoS and security implementation, security and QoS monitoring system, integrators for well-known commercial MOM systems (JMS, Tibco's, Reuters, and IBM's MQ Series), and Broker Manager Agent without and with optimization.

These GEMOM results are being validated in five case studies – a collaborative business portal, a dynamic linked exchange, a financial market data delivery system, a dynamic road management system, and a banking scenario (money transfer). The results of the five case studies will enable us to foresee how the GEMOM system as a whole will perform in different real-life scenarios.

VI. Related work

MOM platforms are available in a wide range of implementations such as JMS, WebSphereMQ, TIBCO, Herald, Hermes, SIENA, Gryphon, JEDI and REBECCA where each of these MOMs has been designed to achieve specific goals, and employs unique functionality to meet specific messaging challenges [27]. However, the current state-of-the-art technologies do not allow security mechanisms to actually predict or anticipate future threats, and to adapt to rapidly changing behaviors and threats over time. There are some areas of research that are promising in this regard.

A number of adaptive security systems have been developed recently supporting adaptation at different levels (from hardware-level to application-level) and for a number of reasons. That is, security in an autonomic computing environment [3], adaptive security for wireless networks [5] and complex information systems [19], adaptable security manager for real-time transactions [11], dynamic authentication for networked applications [18], adaptive firewall architecture [10], threat-adaptive security policy [21], self-contained object for secure information distribution systems [2], adaptive trust negotiation and access control [14], adaptive security policies [7], a bio-inspired self-protecting organic message-oriented middleware [13], anomaly-based self-protection against network attacks [24], and virtualized trusted computing platform for adaptive security enforcement [26]. Several taxonomies have been introduced for classifying adaptive and reconfigurable systems [12]. A survey of approaches to adaptive application security, and adaptive middleware can also be found in [4] and [15], respectively. A bus-based architecture for integrating security middleware services is proposed in [25]. Presentations of semantic and logical foundations of and local and global requirements in an adaptive security infrastructure can be found in [9], [23]. It was the work of the above researchers that convinced us of the viability of adaptive security and trust, and therefore confidence in the productivity of our research in these directions.

Weise [22] presents a security architecture and adaptive security, and discusses a new perspective on the characteristics of a security architecture that is capable of reducing threats and anticipating threats before they are manifested. This architecture is similar to our AES, but our AES goes further by the integration of a continuous cycle of monitoring, assessment and evolution, and tools and processes for pre-emptive vulnerability testing and updating. The architecture is similar to ours in that it uses biological and eco-system metaphors to provide interesting parallels for adjusting and responding to constantly emerging and changing threats, but ours goes further by combining a compromised-based trust model to maximizing the value of risk-taking.

In [20], an active trust management for autonomous adaptive survivable systems is described. The trust model described there forms the basis for our compromised-based trust model. However ours goes further by combining this model with a security-based trust model using an adaptive control loop to minimizing the rate and severity of compromises via the provision of a secure communication environment.

VII. Conclusion and future perspectives

In this paper we have described GEMOM, which provides solutions to overcome limitations in robustness, resilience, adaptability and scalability, and have presented an AES and an ATM approach to such autonomous MOM systems, an approach that is capable of maintaining the proper balance between security and trust, and performance in rapidly changing environments.

Our investigations convince us that our AES and ATM models of GEMOM are capable of impacting on robust communications among users with disparate devices and networks, secure self-healing systems that support mission-critical communication under highly dynamic environmental conditions, self-auditing systems that can report state inconsistencies, incorrect or improper use of components, systematic secure evolution of legacy software to accommodate new technologies and adapt to new environments, and of enabling systems to operate in the face of failures and attacks.

In our future work we plan to prototype those components we have as yet not dealt with, and continue to validate our results in our five scenarios, which are collaborative business portal, dynamic linked exchange, financial market data delivery, dynamic road management system, and banking for money transfers.

Acknowledgment

This research is in the context of the EU project GEMOM (Genetic Message-Oriented Secure Middleware), grant agreement no 215327, approved by the EU Commission. The author acknowledges the efforts of all partners who participated in the development of the GEMOM system. The author would like to thank Wolfgang Leister and Reijo Savola for reading the draft of this paper and for their helpful comments that improved the paper greatly.

REFERENCES

[1] H. Abie, I. Dattani, M. Novkovic, J. Bigham, S. Topham, R. Savola, GEMOM – Significant and measurable progress beyond the state of the art, ICSNC 2008, Sliema, Malta, October 26-31, 2008, pp. 191-196.

[2] H. Abie, P. Spilling, and B. Foyn, Rights-carrying and self-enforcing information objects for information distribution systems, 6th Int. Conference, ICICS 2004, Malaga, Spain, LNCS 3269(0302-9743):546–561, October 27-29, 2004.

[3] D. M. Chess, C. C. Palmer, and S. R. White, Security in an autonomic computing environment, IBM Systems Journal, Vol. 42, No 1, 2003 107-118.

[4] A. Elkhodary and J. Whittle, A Survey of Approaches to Adaptive Application Security, Int. Workshop on Software Engineering for Adaptive and Self-Managing Systems (SEAMS '07), 20-26 May 2007

[5] C. T. R. Hager, Context aware and adaptive security for wireless networks, PhD thesis, Virginia Polytechnic Institute and State University, Blacksburg, Virginia, November, 2004.

[6] IBM White Paper Autonomic Computing, an architectural blueprint for autonomic computing, Third Edition, June 2005.

[7] P. Lamanna, Adaptive security policies enforced by software dynamic translation, A Thesis in TCC 402 25 March, 2002.

[8] S. A. Macskassy and F. Provost, A brief survey of machine learning methods for classification in networked data and an application to suspicion scoring, Statistical Network Analysis: Models, Issues, and New Directions, LNCS Vol. 4503/2007, April 12, 2008.

[9] L. Marcus, Semantics of static, adaptive, and incremental security policies, 1st Symposium on Requirements Engineering for Information Security (SREIS), March 2001.

[10] J. Zou, K. Lu, and Z. Jin, Architecture and fuzzy adaptive security algorithm in intelligent firewall, in Proc. MILCOM, 2:1145–1149, October 7-10, 2002.

[11] S. H. Son, R. Zimmerman, and J. Hansson. An adaptable security manager for real-time transactions, In Proc. 12th Euromicro Conference on Real-Time Systems, pp 63-70, June 19-21, 2000.

[12] P. McKinley, S. Sadjadi, E. Kasten, and B. Cheng, A taxonomy of compositional adaptation, May, 2004.

[13] A. Pietzowski, B. Satzger, W. Trumler, and T. Ungerer, A Bio-inspired Approach for Self-protecting an Organic Middleware with Artificial Antibodies, Self-Organizing Systems, LNCS Vol. 4124/2006, September 21, 2006.

[14] T. Ryutov, L. Zhou, C. Neuman, T. Leithead, and K. Seamons, Adaptive trust negotiation and access control. In Proc. 10th ACM symposium on Access control models and technologies, pp 139–146, June 1-3, 2005.

[15] S. Sadjadi, A Survey of Adaptive Middleware, Technical Report MSU-CSE-03-35, Computer Science and Engineering, Michigan State University, East Lansing, Michigan, December 2003.

[16] R. Savola and H. Abie, On-Line and Off-Line Security Measurement Framework for Mobile Ad Hoc Networks, Journal of Networks Special Issue on Security of Wireless Communication Systems, Academy Publisher, in press, due in November 2009.

[17] R. Savola and H. Abie, Identification of Basic Measurable Security Components for a Distributed Messaging System, The 3rd Int. Conference on Emerging Security Information, Systems and Technologies (SECURWARE) 2009, June 18-23, 2009.

[18] P. A. Schneck and K. Schwan, Dynamic authentication for high-performance networked applications, In Proc. 6th Int. Workshop on Quality of Service, pp. 127–136, May 18-20, 1998.

[19] A. Shnitko, Adaptive security in complex information systems, in Proc. 7th Korea-Russia Int. Symposium on Science and Technology, (8023863):206-210, 28 June - 6 July, 2003.

[20] H. Shrobe, J. Doyle, and P. Szolovits, Active trust management for autonomous adaptive survivable systems, pp. 40-49, Self-Adaptive Software, 2000.

[21] R. M. Venkatesan and S. Bhattacharya, Threat-adaptive security policy, in Proc. IEEE Int. Performance, Computing, and Communications Conference, pp. 525-531, February 5-7, 1997.

[22] J. Weise, Security architecture and adaptive security, SSA Journal, pp. 10–15, July, 2008.

[23] L. Marcus, Local and global requirements in an adaptive security infrastructure, Int. Workshop on Requirements for High Assurance Systems (RHAS 2003), Sept. 2003.

[24] G. Qu and S. Hariri, Anomaly-Based Self-Protection against Network Attacks, in Autonomic Computing: Concepts, Infrastructure, and Applications, Ed.: M. Parashar and S. Hariri, CRC Press, 2007, pp. 493-521.

[25] T. Goovaerts, B. D. Win, and W. Joosen, A comparison of two approaches for achieving flexible and adaptive security middleware, Proc. of the 2008 workshop on Middleware Security, Leuven, Belgium, December 2, 2008. ACM 2008, pp.19-24.

[26] I. Djordjevic, S.K. Nair, and T. Dimitrakos, Virtualised Trusted Computing Platform for Adaptive Security Enforcement of Web Services Interactions, IEEE Int. Conference on Web Services (ICWS 2007), 9-13 July 2007, pp 615-622.

[27] E. Curry, D. Chambers, and G. Lyons, Extending Message-Oriented Middleware Using Interception, Proc. 3rd Int'l Workshop on Distributed Event-Based Systems (DEBS 04), 2004, pp. 32–37.