MONITORING DISTRIBUTED SYSTEMS FOR ADAPTIVE SECURITY

by

Mekonnen Feyissa

A Thesis Submitted to the School of Graduate Studies of Addis Ababa University
in partial fulfillment of the requirements for the degree of
Master of Science Degree in Computer Science

July, 2007

Addis Ababa

ACKNOWLEDGMENTS

I can not possibly thank enough my supervisors, Dr. Demissie B. Aredo and Dr. Mulugeta Libsie, for their guidance and support through the tortuous and challenging path of this thesis work. I am deeply indebted to them for a lot of long discussions and feedback throughout the development of this work and particularly for the efforts made by Dr. Demissie B. Aredo for the success of this work mainly based on virtual discussion. Many thanks should go to all faculty members and colleagues in the Department of Computer Science, Addis Ababa University for their support from time-to-time.

I would like to thank my friends Mr. Belay Mebrat and Mr. Tewodros Nigussie for devoting their precious time to proof-reading drafts of this thesis and for their invaluable comments and constructive criticisms.

Last, but not least, I would like to thank my family, my wife Mrs. Emebet Alemu, my children Hawani, Naol and Yosan for their moral support and their patience during my absence.

Table of Contents

List of Figures

List of Tables

Definitions and Abbreviations

Target monitoring system is a monitoring system attached to each node in the distributed system.

Target monitored system is a system that is monitored by the local (target) monitoring system. It can be on of the computing devices in the distributed system.

Central monitoring system is a monitoring system that controls changes in the distributed system and reconfigures the system based on the changes.

Critical information is information that is highly confidential.

DAMS is an abbreviation for distributed adaptive monitoring system.

GAMS is an abbreviation for global adaptive monitoring system.

ADS is an abbreviation for adaptive distributed system.

EA is an abbreviation for Event Analysis and is to mean a module that performs analysis on the information gathered for adaptation purpose to determine the level of risk the target system is.

GEA is an abbreviation for Global Event Analysis and its function is similar to EA.

ER is an abbreviation for Event Recorder and is to mean a module that listens to a process in execution and triggers an event for the changes occurred.

FIFO is an abbreviation for first in first out.

Critical Time is the time when highly confidential information is transferred to the central monitoring system.

MV stands for metrics value.

Abstract

In today's Information Society, distributed systems have a significant impact on how information communication between social, industrial, governmental and non governmental institutions is achieved. Dealing with the complexity, heterogeneity and dynamicity of distributed systems is absolutely among the major concerns of the software industry. In the Internet era, the distribution of information and services on different sites is a common and dominant scenario. As information is the key tool for any institution's goal achievement, accessing information and services on remote sites over unsecured communication line requires high-level of quality of services (QoS): e.g. an acceptable response time and security mechanisms. Such QoSs require inherent adaptation of the system to changes in the environment, i.e. securing key information while adapting the distributed system.

In this thesis, we have proposed a framework for assessing and determining to what extent we can allow distributed systems to gather information for the purpose of adaptation. We have proposed a dynamic security metrics that is used to measure how far the security of the adaptive distributed system can be affected by a monitoring for adaptation and the necessary measures that need to be taken by the monitoring system. In addition to this, we also tried to discuss the role of a monitoring system in securing the adaptive distributed system and propose an adaptive monitoring system, a monitoring system that adjusts its monitoring functionality depending on the changes in its environment. Adaptive monitoring system allows the adaptive distributed system to gather detailed information based on the changes in the environment and choose efficient mechanisms (algorithms and/or encryption techniques, and secured and cost effective communication channel) for exchanging the gathered information between the target distributed systems and the central monitoring system.

Keywords: Adaptation, Adapting distributed system, Security, Adaptive security, monitoring system, Adaptive monitoring system.

1. Introduction

1.1. Background

An Adaptive Distributed System (ADS) is a system that dynamically adjusts its behavior based on changes in the environment. These changes could be process or link failures, changes in communication patterns or frequency, changes in failure rates, or changed user requirements. Adaptation technique allows software or the system to modify its own functions and configuration in response to changes in its environment. Hence, adaptive distributed systems better know what is happening in their environment by detecting and evaluating the changes in the environment and adjusting their actions to the changes than the non-adaptive ones.

Adaptation can be achieved by allowing the system to collect detailed information by its monitoring subsystem. However, the more detailed information the adaptive distributed system gathers through its monitoring subsystem and other components, the more risky it becomes since an intruder acts more severely in a distributed environment if it takes control over the monitoring component [6].

Allowing the monitoring system to gather information for the purpose of adaptation allows the adaptive distributed system to know more about changes in its environment and as a result the system can have best adaptive capability to take proper action for the changes.

On one hand, allowing the monitoring system to gather more detailed information can cause considerable security problem if the monitoring system is overtaken by intruders and as a result the information is available for the intruders. On the other hand, restricting the monitoring system not to gather more information reduces the adaptive capability of the system. For this reason we have identified the level to which we allow the monitoring system to gather information for the purpose of adaptation measured by the security metrics based on user requirement.

Threats are potential violation of security. A distributed system is subject to security threats whether it is adaptive or not. Some of the security threats in distributed systems are Interruption, Inspection, Modification and Fabrication of information. Hence a secured distributed system should guarantee availability, confidentiality and integrity of information.

Security metrics are expected to measure the level of the security risk based on the attributes of information relevant to security issues. These attributes according to [6] include level of criticality, detail, size and support for inference. The adaptive security metrics discussed in this thesis have the capability to be configured in such away that it dynamically takes the attributes of the information asset that can vary from time to time based on users' interest.

Security metrics indicate the degree to which security goals such as data confidentiality are met. They propose actions that should be taken to improve the overall security program, and identify the level of risks in not taking a given action and hence provide guidance in prioritizing the actions [6]. They also indicate the effectiveness of various components of a security program [4].

Since the information gathered by a distributed monitoring system may be critical, we have considered an adaptive monitoring system that changes its behavior depending on the criticality of the information collected for the adaptation purpose. Hence, the proposed adaptive distributed monitoring system chooses the appropriate action to be taken when the information gathered is security critical. For example, it may allow the monitoring system to collect detailed information when the effective security mechanism of the target system is excellent; allow using encryption mechanism to send the gathered information from the target monitoring system to the central monitoring system when the effective security metrics is above the criticality level.

A security metrics that identifies the level of risk and, that facilitates to take different decisions based on the changes in its environment is an adaptive one. Hence, we need to investigate for adaptive security metrics whose attribute may change dynamically

based on the importance of the information asset. The importance of an information asset may increase or decrease from time to time depending on the situation of the organization. Hence, the weight of information criticality has to change dynamically from time to time based on the importance of the information.

1.2. Motivation

Currently, the development of Information Technology is tending towards the distribution of information rather than centralization. This leads to the need of having a distributed system that has a significant impact on how communication between social, industrial and governmental institutions is achieved. Hence dealing with the complexity, heterogeneity and dynamics of distributed systems is absolutely among the main concerns of the software industry [6]. Accessing information on a remote site requires high-level of system quality: acceptable response time ("Near real-time"); and security mechanisms which in turn require inherent adaptation of the system to changes in the environment. Hence the need for studying adaptive distributed system.

The results from adaptive behavior research are exploited for building artificially intelligent adaptive systems. In this thesis, we believe that the research progress in adaptive behavior will affect the research in adaptive distributed systems. That is, monitoring, change detection and behavior adaptation components of an adaptive distributed system will become more intelligent in time.

Security in any information system, whether adaptive or non-adaptive, has become a serious concern. Currently, the world is full of intruders and several types of security threats are occurring during communication and in the form of unauthorized attempts to access stored information. The monitoring component of adaptive distributed systems can be an external subsystem which is highly vulnerable. Considering the security threats that may occur in the case that an intruder takes control over the monitoring system, we argue that there must be a limit to the kind of information, i.e. its level of criticality, and level of detail of the knowledge that the monitoring system is allowed to have. The more critical information the monitoring subsystem logs, the higher the risk of the security mechanism be compromised. For instance, if the purpose of the monitoring is to provide a better security mechanism by making the system adaptive to changes in the environment, e.g. by capturing

intrusion attempts, the monitoring system should collect detailed and security-critical data such as user ID and IP address of remote sites. In this case, there is a high risk of information disclosure to unauthorized intruders. Hence, establishing a technique for finding a trade-off between collecting critical data to achieve a better adaptation and the risk of running into security threat is necessary. These are among the research issues that motivate us to closely investigate the problem and to propose a concrete solution

1.3. Problem Statement

It is obvious that a monitoring component aims at becoming more knowledgeable about the environment it is functioning in so that the changes in the distributed environment can be detected and corresponding actions can be taken in order to compensate for the changes in the environment for the purposes of providing acceptable quality of service.

There are two main problems in adaptive distributed systems. On one hand, monitoring a system and collecting data necessary for adaptation may cause security problems. Information about activities of users, their communication patterns as well as contents of messages are collected by the monitoring system, which is usually external to the target system. It causes a considerable security threat if the monitoring system is taken over by an intruder and as a result the collected information becomes available to the intruder. The situation becomes more critical as the techniques and mechanisms of gathering and analyzing the collected information become more intelligent in time. Hence, there must be a limit to the kind and the level of detail of the knowledge that the monitoring system can be allowed to have. Moreover, the monitoring system should also be protected from intruders.

On the other hand, restricting the monitoring and gathering of information may constrain the capacity of the system to adapt to the changing environment and maintain the security mechanism. Hence, in making a critical distributed system adaptive to deal with security threats, there may be risk of compromising the whole security mechanism.

Therefore, there must be an investigation on the level of knowledge about a distributed environment. Basically, it is this knowledge that might be required to gather information necessary for adaptation purpose. How this knowledge can be exploited by intruders to cause a security threat is also an area to be focused on. In this thesis, the following core issues will be investigated:

- What is the level of detail of knowledge about a distributed environment which might be required by monitoring systems to gather information necessary for adaptation purpose?
- Formulate an adaptive distributed monitoring system that can adjust its functionality based on the changes in its environment.
- How this knowledge can be exploited by intruders to cause security threats in the situations like Token ring?
- Define security metrics which itself may be adaptive, in order to enable to measure security levels of the adaptive distributed environment.
- Formulation and implementation of a framework for adaptive distributed monitoring system using security metrics in the context of an adaptive distributed system.

1.4. Objective

As discussed above, adaptive distributed systems are systems that can evolve their behavior based on the changes in their environment. To achieve this goal, it is necessary that the adaptive system monitors its environment and collect information pertaining to changes in the environment to be able to evolve its behavior based on the collected data. Hence the general and specific objectives are as follows

General objective

The main objective of this study is to propose techniques and methodologies for quantifying impacts of monitoring subsystems on adaptive distributed systems and to propose metrics that enable us to find the trade-off between collecting detailed information and restricting capacity of adapting to changes.

Specific objective

In particular, our aim is to

- investigate the level of knowledge about the distributed environment that might be required for adaptation purpose and how this knowledge can be exploited by intruders to break into the security mechanism of the target system.
- define the security metrics, which itself could be adaptive, to be able to measure security levels.
- formulating a framework for security metrics in the context of adaptive distributed systems is also among the objectives that we are planning to achieve in this work.

1.5. Scope

This thesis deals with measuring the criticality of information gathered for the purpose of adaptation and the impact of monitoring for adaptation on the security of the target system.

1.6. Methodology

In order to achieve the objectives proposed above and to answer the questions posed in the problem statement, the following activities will be undertaken.

1. We believe that even though we have seen some of them in the literature review it is critical to review the prior works in the area of adaptive distributed systems. Hence we review in detail literature that will help us to understand the problem area.
2. In our problem section, we have indicated that level of knowledge about distributed environment can be exploited by intruders to cause security threats, and hence in this work we will try to identify the type of knowledge that will be employed by the monitoring system which is optimal in such away that the level of details gathered by the monitoring system is not exploited by intruders which is the extension of [6]

3. In addition to this there is security metrics defined in the previous work [6]. Hence we will try to refine and adapt the existing security mechanisms and security metrics in the context of adaptive distributed system.
4. We also believe that to reach on to the goals that we have aimed, laboratory testing in the real environment and/or simulated environment for adaptive distributed systems is vital. Hence it might be necessary to collect data and test it in a simulated adaptive distributed environment whether we have successfully investigated the thought goal.

1.7. Thesis organization

The rest of the thesis is organized as follows: In Section 2, a brief problem statement is presented. In Section 3 reviews of literature around adaptive distributed systems and security metrics are summarized. In Section 4, we propose a method on how to determine the level at which we can allow the monitoring system to gather information for adaptation purpose, an adaptive security metrics, adaptive distributed monitoring system and its model that measures the level of security risk of the target system dynamically. In Section 5, we discuss the implementation approach and finally conclude the thesis by summarizing and discussing the potential research issues for future work in Section 6.

2. Literature Review

2.1. Adaptation

Biological organisms adapt to their environment in order to get air, water, food, nutrients and to protect themselves from their enemies [22]. In order to cope with physical conditions such as temperature, light and heat; to defend themselves from their enemies; and to respond to changes around them, organisms adapt themselves to the change. Adaptation enables living organisms to cope with environmental stress and pressures. Such adaptation can be structural or behavioral. Structural adaptations are special body parts of an organism that helps it to survive in its natural habitat such as skin color, shape, and body covering. Behavioral adaptations are special ways in which a particular organism behaves to survive in its natural habitat. Organisms that

cannot suitably adapt to their environment will either have to move out of the habitat or die out.

Likewise, distributed systems can be perceived as living organisms in the sense that the state of the computing system as well as its execution environment conditions change dynamically. In order to provide the intended services and functionalities at the required quality of service, adaptation of the system to the changing execution environment is necessary. Adaptive systems that can change their behavior at run time have a number of potential benefits. Fore example, adaptive distributed systems can respond rapidly to security threats to improve the opportunities to optimize performance as the execution environment changes.

Therefore, it is possible to adapt a computing system to any change in its environment to provide the intended functionalities as living organisms do in their habitat. This can be achieved by allowing the monitoring system to collect different parameters dynamically at run time about the changes. Allowing the monitoring system to collect detailed information for the purpose of adaptation increases the adaptability capability of the target system.

On the other hand, allowing the monitoring system to gather more detailed information for the purpose of adaptation may cause serious security problems if the monitoring system is overtaken by an intruder and as a result the information is available for a third party. We argue that there must be a level to which we can allow the monitoring system to collect information for the purpose of adaptation. This thesis tries to determine this gap, the level to which we can allow the monitoring system, by measuring the security-criticality of the information gathered using security metrics and decide the way to communicate these information between the distributed monitoring system and the central monitoring system so as to minimize the security threats of the distributed system in general and the monitoring system and the communication channel in particular. As a biological organism uses adaptation to escape from its enemies, an adaptive distributed system can also use adaptation to control malicious activities at the process creation level. Hence, in this thesis we try to

make the distributed monitoring system adaptive by adding extra layer or component to the distributed monitoring architecture proposed by Tsai *et al.* [21].

2.2. Adaptive Distributed Systems

An adaptive distributed system is a system that changes its functionalities at run time in order to adapt to variation in the execution environment such as resource availability and user mobility. Silva *et al.* [1] argued that component migration, replication and reconfiguration are essential to support user mobility and to cope with resource availability fluctuations. Hence, they have developed an object-oriented model that is composed of three packages namely, monitoring package, event detection and notification package and dynamic reconfiguration package. The Monitoring package is responsible for monitoring the entities present in the system such as process utilization in the various hosts of the distributed system and informs the event detection and notification package whenever the value of the parameters collected significantly changed. The event detection and notification package is responsible for analyzing data and determining the occurrence of events. The dynamic reconfiguration package is the one that takes the required reconfiguration action to adapt the system.

Pal *et al.* [2] described that current advances in distributed systems technology are giving rise to a new breed of sophisticated systems that are flexible in their configuration and agile in their behavior. Such a system monitors and responds to changes in environmental conditions by altering its configuration and behavior automatically to provide the best quality of service possible. They have focused their research on the timeliness of the auto-Adaptive distributed system in case there is a malicious attack to the system. They strongly argue that if the system adapts too late, it may not be useful and may even be detrimental in the cause of survivability that involves defensive reactions such as blocking a port which is the source of attack packets or killing a process when there is a maliciously started process or removing a file if there is a Trojan horse. Hence, they have associated the two factors in adaptive behavior in distributed systems namely how quality of service is offered by the

system and how to respond to malicious manipulation of the environment by attackers, i.e., survival and defense.

Hiltunen and Schlichting [3] formulated a model for adaptive distributed systems that divides the adaptation process into three phases namely Change detection, Agreement and Action. Change detection is a component that monitors a possible change in the environment and decides when to suspect that change has actually occurred. Agreement is responsible for reaching an agreement among all sites that adaptation is required and action phase is responsible for changing the behavior of the system based on the agreed changes. From the agreement phase we can easily formulate that the agreement is done between different systems that are located on different sites and the agreement is reached by sending information between the sites that are susceptible for intruders. The authors consider different examples where Distributed Adaptation is required. Based on this they have considered reliability protocols for message passing, and adaptive concurrency control. They have also discussed and assessed different examples of adaptive algorithms where the change in the environment is processor or communication link failure such as change in group membership, loss of token in a token based network and centralized and total ordering plus detection of server failure.

Chen *et al.* [4] described architecture for developing adaptive software for a distributed system based on cactus, a system for constructing highly configurable distributed services and protocols, software that can change its behavior at runtime. Such software has a number of potential benefits, ranging from the ability to respond rapidly to security threats to the opportunity to optimize performance as characteristics of the underlying execution environment changes. The model of the software system architecture is distributed in such a way that each host consists of multiple components, adaptive or non-adaptive, organized into layers. The adaptive component controls the component's adaptive behavior and alternative adaptation aware algorithm modules that provide a different algorithm that implements the functionality of the component.

A monitoring system is a system that is used to monitor a target program's execution. The monitoring system usually consists of a monitoring module and, monitoring hardware and is used to identify events, detect events and process events. Monitoring is accomplished in two operations namely triggering, detection of predefined events during program execution that activates recording of data pertinent to the event, and recording which is collection and storing of data pertinent to the event. Tsai et al. [21] described the need to have a distributed monitoring system and proposed a distributed monitoring architecture. The monitoring system is assumed to be integrated to each client in the target monitored system in order to avoid high perturbation of the system's data paths. Each monitoring system detects events of interest, an event created as a result of either malicious act in a target monitored system or information gathering for the purpose of adaptation, and records data generated by the events of interest. Each target monitoring system uses a separate network from the distributed network to avoid network overload by the transmission of monitoring information to the central monitoring system. According to Tsai *et al.* [21], the monitoring system can be hardware, software or hybrid. "Hybrid monitoring is an attractive compromise between intrusive software monitoring and expensive non-intrusive hardware monitoring". The hybrid monitoring minimizes perturbation by allowing hardware to perform the majority of the monitoring activities.

2.3. Security Threat and Metrics in Adaptive Distributed Systems

Aredo and Yildirim [6, 12] described that there must be security metrics, which is a tool designed to facilitate decision-making and improve the performance and accountability through collection, analysis, and reporting of relevant performance related data. In order to quantify the impact of monitoring on the effectiveness of a security mechanism of the target system, they defined some metrics as a function of sets of attributes of data to be collected by the monitoring system. The attributes that are relevant to security issues are the level of criticality, detail, size and support for interference. Hence the security metrics, M, can be defined by the equation

$$M = \alpha.C + \beta.D + \lambda.S + \eta.I \qquad (1)[6]$$

Where

- C is the level of security criticality attribute of the data
- D is the level of the detail of the data
- S is the size of the data
- I is the support for inference and
- α, β, λ and η are non-negative coefficients whose values and relationships can be determined using some analytical techniques.

Since security metrics and the effectiveness of the security mechanism of the target system are inversely proportional [6], a security mechanism can be given as

$$SM = {}^{1}\!/_{M} \quad \text{.......................................} \quad (2)$$

Where SM is the effectiveness of the Security Mechanism of the target system.

Voas *et al.* [7] described a software assessment method which they call it Adaptive Vulnerability Analysis (AVA) that measures a program's relative security weakness in terms of known and unknown threats that may occur in the future. Their methodology is based on the measurement of security weakness in terms of previously known threats. As a result, the resulting metrics vary with different set of threats and hence adaptive. The advantage is its ability to be customized to application specific classes of intrusion. The fact that it is based on a predefined set of threats is its limitation. Even though they stated that AVA measures the software's security based on known and unknown threats AVA fails to account for clever intruders who create new malicious threats from scratch [8]. Hence it totally provides relative measure based on previously known threats.

Swanson *et al.* [12] assessed elements that must be considered in defining effective security metrics like metrics must yield quantifiable information; supporting data must be readily obtainable; only a repeatable process should be considered for measurement; and metrics must enable tracking of performance.

Information attacks can be on transit or on the stored information as an unauthorized access. Langweg and Snekkenes [8] present a classification of attacks by malicious

software, which focuses on application software rather than operating systems. The classification is based on three dimensions namely location of a vulnerability, a place where fault can be introduced, cause of vulnerability and manifestation of security property violation. Foundstone strategic security [14] also tried to discuss attacks of information on transit and proposed how much risky a network system is.

Chidamber and Kemere [9] developed a software metrics for object oriented design based on measurement theory that provides senior designers and managers who may not be completely familiar with the design of details of an application. Such metrics is used to identify areas of the application that may require more rigorous testing and areas that are candidates for redesign.

Salsano *et al.* [10] discussed how to secure messages on the session initiation protocol (SIP), an Internet Engineering Task Force (IETF) standard for IP telephony. Since SIP messages may contain information that a user or server whishes to keep private, end-to-end or hop-to-hop authentication procedure is implemented. For example, the header can reveal information about the communication patterns and contents of individuals, or other confidential information. Hence security of such SIP information is a critical issue. The two security mechanisms used in SIP are authentication and data encryption that are used to authenticate the sender of the message and to ensure that some critical message information was unmodified by attackers in transit.

2.4. How to Adapt a Distributed System

Recent advances in distributed systems technology are giving rise to a new breed of sophisticated systems that are flexible in their configuration and responsive in their behavior. They have the capability to monitor and respond to changes in their environmental conditions by altering their configuration or their behavior in an effort to provide the best quality of service possible such as defending against malicious attacks.

"Adaptation techniques allow a distributed system to modify its own functions and configuration in response to changes in its execution environment. The changes might

include, for example, processor or link failures, changes in communication patterns or frequency, changes in failure rates, or changed user requirements" [1]. Adaptation requires changing the computing system at runtime dynamically. Different researchers proposed a framework for adaptive distributed systems that can be applied in many scenarios. The model proposed in [3] is composed of three main packages namely the monitoring package, event detection and notification package and dynamic reconfiguration package in response to the changes in the environment.

The monitoring phase of the adaptive distributed system is expected to gather information about the status of the distributed resource such as memory, CPU, disk and network connectivity [5]. The information gathered by the monitoring component for the adaptation purpose dynamically during run time is either stored in a log file or directly sent to the event detection and notification package where it can be analyzed and determine the events that are relevant to the application.

To implement the monitoring functionality, Silva and Edler [1] used a resource monitoring object, which is responsible for monitoring system specific parameters such as processor utilization and notify the event detection and notification package whenever the value of the parameters change significantly. In addition to the resource monitoring object, modern distributed systems also use the concept of interceptors as defined in CORBA [15]. Interceptors are inserted into the object invocation path and each time a client invokes the method of an object, the message corresponding to this invocation is intercepted and later re-dispatched to the target object. Using interceptors, the system can extract useful information from each method invocation, storing it in a log file for analysis by the event detection and notification package. This method is highly vulnerable to security threat if such a system is external and overtaken by intruders.

The dynamic configuration package or simply the configuration component of the adaptive distributed system is responsible to take the proper action depending on the outcome of the event detection and notification package analysis. The algorithm from which the adaptive system is constructed is expected to have different execution paths. The configuration component selects from these execution paths depending on

the adaptive action that may include but not limited to this adjusting parameters or any event trigger it will take [5].

Potential benefits of Adaptive Distributed systems (ADSs) include the ability to respond rapidly to security threats, reliable message transmission, consistent messages ordering across hosts, implementing functions such as replication or data consistency for higher level services such as a network directory service and the opportunity to optimize performance as changes in the execution environments take place [1]. In addition to the above benefits, adaptive distributed model can be applied for achieving timeliness in the construction of fault tolerant ADS as applied by Pal and Hiltunen [2, 3].

2.5. Summary

Different authors have discussed how distributed systems can change their behavior and functionalities at run time to adapt to changes in their environment, and respond to the changes gathered through their monitoring system as compared to their equivalent non-adaptive ones. To develop such agile systems, called adaptive distributed systems, the researches developed an adaptive distributed systems model that divides the adaptation process into three phases. From the literature review, we can see that as we allow the distributed system to gather more detailed information for the purpose of adaptation through their monitoring system, the system will have more capability for adaptation. On the other hand, as we allow the monitoring system to gather more detailed information, there is a security problem if the monitoring system is overtaken by an intruder. Different researchers have [6, [11] identified such a problem and agreed that there must be a limit to which we can allow the monitoring system to gather information. But they didn't determine the level and didn't discuss how to measure this level. What measures to be taken by adaptive distributed systems when the monitoring system tries to gather beyond the limited level set is still an open problem.

Security issues in distributed systems, whether adaptive or non-adaptive, are serious. But none of the authors discussed adaptive security metrics, a security metrics that

proposes the necessary measure to be taken based on changes in its environment that measures the degree to which security goals are met at run time dynamically. Aredo and Yildirim [6] proposed parameters required to define the impact of monitoring on the effectiveness of a security mechanism of the target system. But the need to have adaptive security metrics is an open problem in their work.

Therefore, in this thesis, we would try to fill the gap identified above, i.e., we would try to determine the level to which we can allow the monitoring system to gather detailed information for the purpose of adaptation that can be measured by security metrics and what measures to be taken when the monitoring system tries to gather information for the purpose of adaptation that yields a security metrics value beyond the determined level. In addition to this, we will try to add additional layer or component to the architecture of distributed monitoring system proposed by Tsai *et al.* [21] to make it *adaptive monitoring system* and determine the way to communicate the information between the distributed monitoring system and the central monitoring system and vice-versa dynamically at run time to reduce information interception on the communication channel when security-critical information is going to be exchanged. The information can be determined whether it is security-critical or not based on the adaptive security metrics result proposed in this thesis.

3. Attributes of Monitoring Adaptive Distributed Systems

The basic components of an adaptive distributed system are *monitoring*, *change detection* and *reconfiguration* in response to the changes in the environment. In this section, we will extend the work of Aredo's and Yildirim's that focus on the monitoring component of Adaptive Distributed Systems and elicit possible impacts that the monitoring component may have on security mechanism and possibly determine what should be the level of detail of knowledge about a distributed environment which might be required by monitoring systems to gather necessary information for adaptation purpose.

The monitoring component is employed for collecting information on attributes that can later be analyzed to detect changes in the environment of the target distributed

system so that the necessary action will be taken by the system. The attributes that are relevant to security issues include: level of *criticality*, *level of detail*, size and *support for inference*.

The level of the criticality of the data to be gathered by the monitoring system is an attribute that indicates the importance of the data with respect to security or it is the information that causes privacy violation of an individual, reduce the competitive advantage of a company or that causes damage to an organization if disclosed to an unauthorized party.

The level of detail attribute of information to be gathered shows the level of abstraction/concreteness of the monitored data. The size attribute measures the amount of data collected during monitoring. It indicates the possibility of network congestion in the target system and support for inference attribute is the possibility to derive security relevant information about other data objects from the given data.

Security metrics is defined as a function of the set of those attributes of data gathered by the monitoring system that are relevant to security issues. Let us consider two extreme scenarios (when the value of the security metrics is very small and very large) for the security metrics, whose value is computed by Equation 1. First, consider when the value of the security metrics is very small. When the value of the security metrics is very small, it implies that there is less or no risk to the target system. Ideally this can be achieved either if information relevant to security issue is not available for the intruders or there is no information gathered by the monitoring system for the purpose of adaptation and hence no security risk. If there is no information gathered by the monitoring system it implies that there is no adaptation. Hence no need to consider this case. Let us consider when there is gathered information for the purpose of adaptation and the information is not available for intruders. This leads us to conclude that either the information is protected by some mechanism or the information gathered don't have security problem if intercepted by the-man-in-middle. Let us further consider the two cases we have identified.

Scenario 1: Information is security-critical and protected by some mechanism

Possible ways to secure information over unsecured channel are, among others, to use secure socket layer (SSL) as described in [20] or to use some cryptosystems. If we use SSL or cryptography for all communication between the monitoring system and the target system, we believe that we are adding extra load to the system resulting in slow performance and congested communication of the target system. As a result, if the system adapts too late it may not be useful and may even be deterimental and the adaptation rather than improving, it may worsen the desirable capabilities such as improved reliability and availability, configuration flexibility and performance of the target distributed system as described in [19].

Cryptography is the area of expertise that deals with secret writing or the art or science of storing information in a form which hides it from those not meant to see it. To achieve this, a cryptosystem requires an encryption key. Hence it is only possible to see the encrypted document if only the encryption key is known. There are two types of Cryptography systems namely symmetric or a secrete key /single key/ cryptography or classical cryptography and asymmetric or public key cryptography.

Symmetric cryptography, also known as secrete key or single key cryptography, uses a single secrete key for both encryption and decryption. Where as asymmetric cryptography, also known as public key cryptography, is a form of cryptography in which a user has a pair of cryptographic keys – a public key and a private key. The private key is kept secrete, while the public key may be widely distributed. The keys are related mathematically, but the private key cannot be particularly derived from the public key. A message encrypted with the public key can be decrypted only with the corresponding private key.

Since classical cryptography shares single key between the two parties who want to communicate secrete information, it is not possible to bind an identity to a key. On the other hand, public cryptosystems use two keys, one of which to be available to all. The public key infrastructure (PKI) is a key management system defined by the way it handles certification, distribution, and secure storage and revocation operations of the public key values. Among the operations of the PKI listed above the most difficult

operation is the key revocation since it deals with up to date validity of the public keys.

Classical cryptography or symmetric cryptosystem requires the sender and the recipient to share a common key. Therefore, every user in the target system requires sharing common key with the central monitoring system. Hence, the central monitoring system is expected to store and manage all the keys that it shares with each target system. As the number of target system increases in number, how to communicate the changes if they want to change the common key they have agreed up on and how to communicate the first common key is a problem [13].

Even though public key cryptography reduces the problem of how to communicate the first common key problem seen on classical cryptosystem by using two keys to managing public (vs. secret) key, it uses third party called Certification Authority to certify a given public key that belongs to a certain user. Because of this, liability issue will be raised. For example, the reliability of the electronic evidence provided by the certification authority depends on its operational standards of integrity, and ultimately on its liability exposure. Thus paradoxically, any well-financed entity is better out of the certification authority business. Accordingly, banks should trust themselves as a certification authority, which defeats the economies of scale anticipated from the public key cryptosystem. Hence, considering this in general and the key management is the headache to deal with. Therefore, it is advisable if the adaptive system uses SSL or cryptography only at *critical times*. But this requires determining the critical time dynamically at run time. The critical time that we have mentioned is when the information gathered for adaptation purpose is security-critical and when such information is transferring to the central monitoring system. As we have explained, the security-criticalness of the gathered information is determined by the security metrics of equation 2 defined in section 3.3.

Scenario 2: Information gathered is not security-critical

Most of the time, information gathered by a monitoring system may not be security-critical. This means that the value of the security metrics is small, which in turn shows the information is not critical, may not support inference from information about

other security-critical data about the node in the target system and the information gathered is not detailed. Basically security issues will be raised when the monitoring system gathers security-critical information for the purpose of adaptation. If we assume that the gathered information is not security-critical, ideally this takes us to conclude that the information is either public information which doesn't need security or the information does not have damage if intercepted by other third party. In normal situation in a target distributed systems there are sensitive information as well as non-sensitive information. For this reason, this thesis deals with measuring the criticality of information gathered for the purpose of adaptation and the impact of monitoring for adaptation on the security of the target system. Hence determine whether the information gathered is security-critical or not and propose the necessary adaptation that the monitoring system will take accordingly.

To facilitate communication between the client and the server in object based distributed system, Common Object Broker Architecture (CORBA), implements client and server side object request Broker (ORB), responsible for enabling communication between objects and their clients while hiding issues related to distribution and heterogeneity. Since CORBA's client side software is required to be kept to the minimum, it uses proxies to connect client application to the underlying ORB instead of generating an object-specific proxy in such a way that the client can dynamically invoke objects through the dynamic invocation interface. But this requires client side interface implementation that either instructs the developer to use an interface definition language compiler or provide the client's proxy itself. But this has limitation with CORBA's objective of portability and distribution transparency. The other approach is to forget about object-specific matters and try to rely on the client-side ORB for the necessary support, but this has also limitation that it can't be able to adapt the client side software when needed. CORBA's solution for this problem is to use interceptors, piece of code added to ORB that modifies an invocation request on its way from the client to the server, accordingly adapts the associated response and provide general mechanism to support extensibility. An interceptor in CORBA can be placed either between the client's proxy and the ORB called request-level interceptors or between an ORB and the underlining network called message-level interceptors.

To solve the problem of interoperability problem, CORBA introduces the so called General Inter-ORB Protocol (GIOP) in which a request message of the client contains a complete marshaled invocation request, comprising an object reference, the name of the method that is to be invoked, and all the necessary input parameters. But the use of interceptors can be a mechanism to breaking into the whole distributed system if one of the clients is overtaken by intruders in such a way that the underlying ORB is controlled by the intruders and as a result, the request message is available for them which enable them to act as an interceptor. This shows that in modern distributed systems as discussed above, there is a high probability of the information to be available to intruders and are vulnerable to security threat.

Therefore both implications violate the current situation of adaptive distributed system. Hence, we can conclude that the first scenario that assumes the information gathered is security-critical and protected by some security mechanism can not be true all the times. This is because as we have discussed above, distributed systems are vulnerable to security treats as we can't have hundred percent secured systems by its very nature. Hence the extreme case when the security metrics is very small can not hold always for the adaptive distributed system.

Secondly, let us consider the case where the value of the security metrics is very large leading to a very small value of the effectiveness of the security mechanism. Basically high value of security metrics is obtained when we allow the monitoring system to collect more critical data, at high level of detail, or large size of data, and that implies high value of support for inference. Hence, the more risky our target system is. Therefore, allowing the monitoring system to gather more information for the purpose of adaptation may cause security problems. Then, the real challenge is to find a trade-off between the two scenarios, i.e. monitoring at acceptable values of the four attributes without putting the security mechanism of the target system at risk.

The above two scenarios show that we can't have a system with almost hundred percent effective security mechanism, i.e., very small security metrics value because of various reasons. Moreover, we can't talk of security threat of the adaptive distributed system with almost zero effective security mechanism, i.e., very large security metrics

value. Hence, there must be a security metrics value that compromise the above two extreme case scenarios and indicates to what level we can allow the monitoring system to gather information for the purpose of adaptation. According to the scenarios we have discussed, the security metrics value that compromise the two extreme cases must yield the level to which we can allow the monitoring system to gather information for the purpose of adaptation while achieving the required quality of service expected from the target system. In other words security metrics value must be a value that yields the tolerable security risk of the target system. Hence we can rewrite Equation 1 as follows

$$M = \alpha.C + \beta.D + \lambda.S + \eta.I \leq K \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (3)$$

Where k is a constant showing the maximum value of security metrics (M). In other words, since M is inversely proportional to the security mechanism, K is the minimum value of the effectiveness security mechanism that the target system can tolerate while delivering the required functionality at acceptable level of quality of service (QoS).

Since the degree of confidentiality of given information varies from person to person or from organization to organization, the level of tolerable security mechanism also varies. Hence, the level to which we allow the monitoring system to gather information for the purpose of adaptation also varies. In general, we can categorize the security mechanism of the target system into five levels: *Excellent*, *Above Average*, *Average*, *Below Average* and *Poor*. Hence a computed security metrics value whose effective security mechanism ranges from 85 to 100 percent can be considered as excellent, 71 to 85 percent as above average, 51 to 70 as average, 26 to 50 as below average and below 26 as poor security mechanism. Table 1 summarizes the range of values of security mechanism and their corresponding efficiency ranking.

Table 1: Effective Security ranking of an adaptive distributed system

Value Range in %	**Ranking**
85 – 100	Excellent
71 – 85	above Average
51 – 70	Average
26 – 50	below Average
0 – 26	Poor

Therefore, we can allow the monitoring system to gather information for the purpose of adaptation that produces a security metrics with tolerable vulnerability whose effective security mechanism is above 85% or more to have excellent security threats protection. The proposed effective security ranking of the adaptive distributed system can be used to determine to what level we can allow the monitoring system to gather information for adaptation purpose. In addition to this, the proposed security ranking must be configurable in such a way that users can vary the rank based on their need. Even though the investigation of the impact of setting the security mechanism to our system is beyond the scope of this thesis, we argue that increasing the security requirement may result in loosing the quality of service we expect from our system.

3.1. Adaptive Security Metrics

"Security metrics is the application of quantitative, statistical, and/or mathematical analyses to measuring security functional costs, benefits, successes, failures, trends and workload" [12]. Metrics can be and are used as individual data points. They are often best used in the depiction of trends. For example: is the cost of security at company X going up or down? The other way of tracking costs and benefits is through the formal project plans. Security functions are "level-of-effort" (LOE), never-ending, daily work, while projects have a beginning and ending date with a specific objective and associated discrete costs.

Metrics can also be used as tools designed to facilitate decision making and improve performance and accountability through collection, analysis, and reporting of relevant performance related data. Hence, Information Technology (IT) security metrics must be based on security performance goals and objectives that state the desired results of a system security implementation. IT security metrics enables us to measure the achievement of the goals by quantifying the level of implementation of security controls and efficiency of the controls, analyzing the adequacy of security mechanism and identifying possible action to address any security lack.

Different researchers described that security metrics provide a practical approach to measuring information security. They are effective tools for evaluating the effectiveness of the components of a security program, the security of a specific system, product or process and the ability of a security department to address security issues for which it is responsible. Even though metrics in software are not direct as in other disciplines, they can help in providing information for decision making on security mattes.

The attributes gathered by the monitoring system have mutual dependencies among themselves, for example, the more the information is detail the more its volume or size. Therefore we can write

$D = \theta.S$ For some non-negative coefficient θ …….. (4)

Substituting for D in equation 1 yields

$M = \alpha.C + \varphi.S + \eta.I$ For φ = β.θ+ λ …… (5)

Where C- is the criticality level of the information gathered that can be equated to the weighted value

S - Size of the information and

I - is support for inference.

The security metrics value determined by Equation 5 above varies depending on the sensitivity of the attributes gathered by the monitoring system for adaptation purpose or based on changes in its environment. To increase the efficiency of the adaptive distributed system, it is not necessary to keep the security metrics always to a value that yields excellent security mechanism of the target system. Hence, the security metrics value varies depending on the changes in its environment or changes in the criticality, detail and inference of the attributes collected for the purpose of adaptation and indicate dynamically the level of risk the distributed system is. Hence, the security metrics indicates different decisions or actions to be taken to secure the target system at different levels. This difference basically originates from the security criticality of events detected. For example, when the security mechanism of the target system is bellow average, it may alert to restrict the monitoring system to gather detail

information for the purpose of adaptation or force the monitoring system to communicate the attribute information gathered through a secured channel such as to use secured socket layer (SSL) or to use other cryptographic system. On the contrary, when the effective security mechanism of the target system is excellent, it may decide to allow the monitoring system to gather more detailed information or allow the distributed system to adapt as much as possible.

The restricting and allowing of the monitoring system can lead us to investigate more on how the monitoring system behaves based on the changes in its environment. We will discuss the adaptiveness of the monitoring system in section 4.2 in detail. The varying values of the security metrics basically originates from the security criticality of the information gathered. The criticality of a given information also varies from time to time based on the importance of the information. Hence, we have designed information criticality matrix that shows the current criticality of the information asset we have in a distributed system (see Table 2). The user can change this information classification based on his/her need at any point in time. Hence, security metrics that computes how much risky the target system is based on the *dynamic information asset* classification and facilitates decision making based on changes in the environment to take the appropriate action is adaptive itself and hence called adaptive security metrics.

Since criticality of information asset varies from time to time, we propose to assign a weighted value (see Table 2) that properly define our need to our information asset in such a way that it shows the degree of criticality of the respective information at a given time. Hence, X1 will be assigned higher weighed value and X3 will be assigned less or almost no security value. If the information gathered by the monitoring system is highly critical, for whatever value of the other attributes (size, detail and support for inference) the proposed adaptive security metrics must yield 85 percent or above effective security mechanism. Therefore, for highly critical information, the security metrics equation can be written as

$$M = \alpha.C + \varphi.S + \eta.I \geq 85\% \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (6)$$

Adaptive security metrics therefore, provide appropriate information dynamically at run time that helps to decide the appropriate measure to be taken at different levels depending on changes in the target distributed system. Such adaptive security metrics knows the appropriate action to be taken to control malicious acts than their equivalent non-adaptive ones.

Table 2: Information Asset Classification table (dynamic)

Information Classification	Weighted Value	Remark
Extremely Critical	X1	Requires 100% security
Critical	X2	Requires 85% Security
Moderate	X3	Requires >50% security
Public	X4	Requires No security

3.2. Adaptive Distributed Monitoring Architecture

Monitoring is the process of extracting information from a target program at run time to reconfigure or adapt the distributed system. From the literature review, the model for adaptation divides the adaptation process into change detection, agreement and action. Among the adaptation process, change detection is implemented by the monitoring system, a system that gathers information for the purpose of adaptation during run-time. Monitoring operation is accomplished based on an event-driven execution paradigm. This implies that clients in a distributed system detect the event as a result of changes in their environment and send information about the changes occurred in a target system to a central monitoring system to reach the agreement that the change had been occurred in the distributed system. This requires the implementation of the adaptation program that is actively running on each of the nodes in the target Distributed System.

In distributed monitoring architecture, to monitor the target distributed system, a monitoring node is connected to the address, data, and control buses of every node of the target distributed system in order to avoid high perturbation of the system's data

paths and each monitoring system detects events of interest on each target monitored system. After identifying the changes, the target monitored system sends the event notified to the central monitoring system to notify other nodes in the distributed system that the change had been recorded. The central monitoring system certifies that the change had been occurred in the distributed system and reconfigures the distributed system in general.

Among the implementation methods of the monitoring system, hybrid monitoring system uses two different triggering approaches: Memory-mapped and coprocessor monitoring. In memory mapped monitoring, a set of pre-defined addresses are used to trigger data recording. The monitoring unit is mapped onto the memory addresses with each address representing an event. In co-processor monitoring approach, the co-processor instructions are used to trigger event recording. The recording unit acts as a co-processor that executes the monitoring instructions. To invoke recording of data pertinent to events of interest, the co-processor instruction is sent by the target processor to the monitoring unit.

The previous work of the monitoring architecture is composed of components namely the *Event Recorder* and the *Bus Interface Module*. Event recorder (ER) component of the monitoring system consists of trigger recognizer and data collector, timer, Overflow counter, control and FIFO Buffer. The main function of the event recorder is to trigger an event when there is a change in the distributed system. This change can be as a result of malicious acts. The timer is used to stamp each event with the current time and is used to record execution time. The overflow counter counts the number of events that have not been recorded due to buffer overflow. Thus the user knows if events were not recorded. In a coprocessor monitoring system, the trigger recognizer and data collector executes the coprocessor instruction sent by the target processor to check whether the events are enabled or not. If the event is not enabled, they continue monitoring events. If the event is enabled, they assemble the keywords of the events, overflow control, and current time together as an event entry to the repository.

Each monitoring node has its own local clock to order the events on the node and to synchronize the time difference caused among each node as a result of the drifting nature of quartz-controlled oscillators, a central monitoring computes the clock difference with respect to each local timer and adjust all local times to the newly calculated time at some specified intervals.

To secure information gathered form the triggered event for the purpose of adaptation, it requires securing the local monitoring system, the communication path between the local monitoring system and the central monitoring system, and requires securing the central monitoring system. To achieve this we have added another additional layer on the existing monitoring architecture namely the *Event Analyst*, *Comparator* and the *Repository (R)* (Figure 1) local to the target monitored system.

Events can be created by the operating system for the healthy operation of the computing system. Hence, we believe that it is necessary to identify events of interest from the events triggered as a result of healthy operation. For this reason, we have considered a repository (R) that contains some of the pre identified events and used to compare events triggered against the one in the repository. Event Analyst (EA) component, see Figure 1, is the component responsible to measure the security-criticality of the data collected by the trigger recognizer and data collector component of the ER locally in the target monitored system. If the data collected is of security-critical, the event analysis decides on how to send the data to the central monitoring system, i.e., either to send it using SSL or using other encryption mechanism or to send it without using SSL or other encryption. Hence, we can secure information on transit by using secured channel.

The monitoring system therefore, checks on the criticality of the information gathered before it sends it to the central monitoring system. Monitoring system that changes its functionality based on the information critically it gathers for adaptation purpose is adaptive itself; hence Adaptive Monitoring System. The comparator is used to identify the event recognized as event of interest or an event created as a result of healthy operation of the target monitored system. This requires the storage of normal events

in a repository (R) (see Figure 1) and comparing the event recognized against events in a repository.

Monitoring system performs its work at process level or function level. Working at process level is advisable because of two reasons. 1) a process is the minimum program unit that can exhibit non-deterministic behavior, and hence, if we can isolate faults to an individual process, we can possibly use the conventional cyclic debugging method for the successive fault isolation levels of abstraction; 2) we can reconstruct the execution behavior for inter-process communication and synchronization operations to localize faults to an individual process

The EA component of the local monitoring system is expected to measure the security-criticality of the data collected by the Trigger Recognizer and Data collector components of the Event Recorder and adjusts its monitoring functionality accordingly. Each local monitoring system communicates the information to the central monitoring system through a separate network in order to reduce perturbation or network traffic.

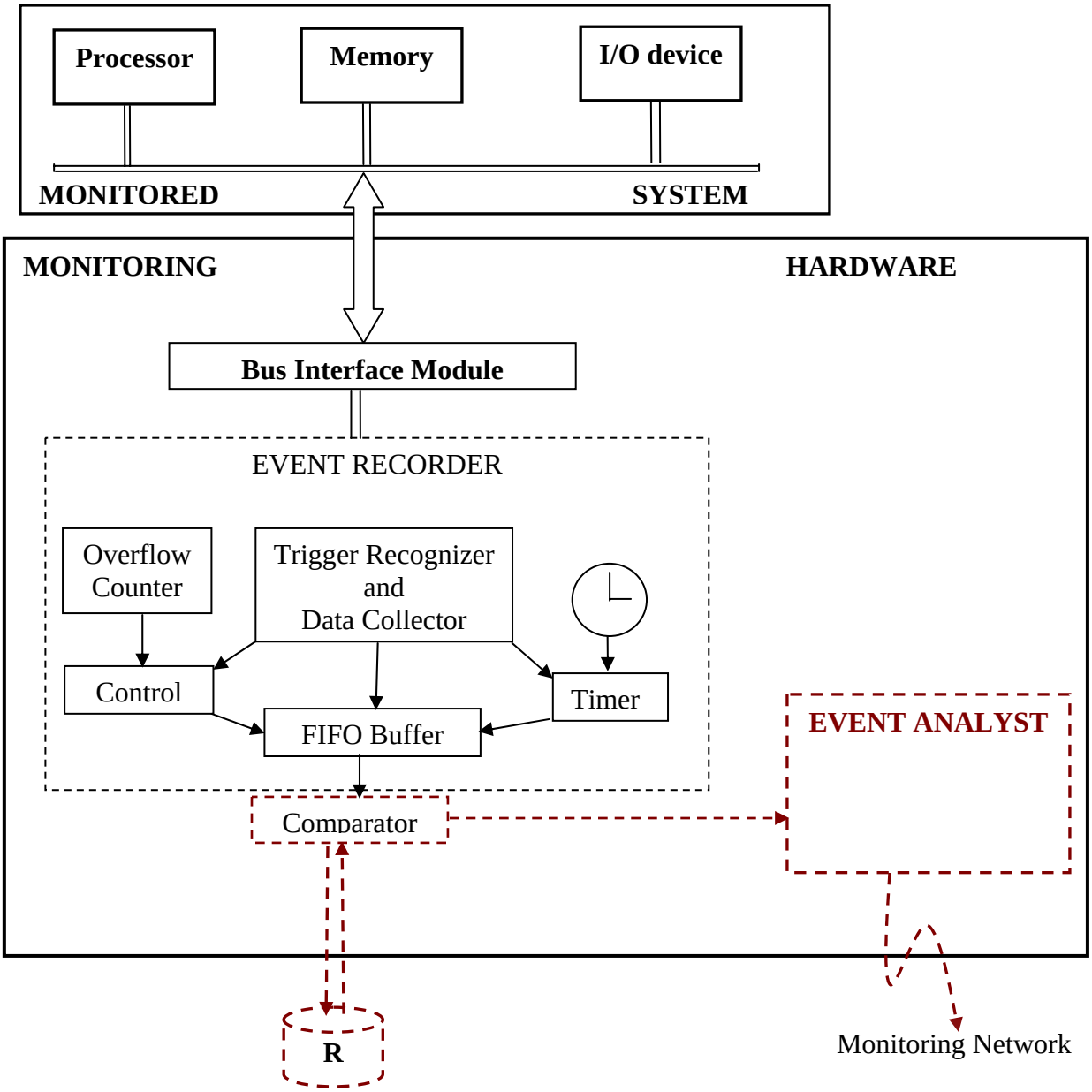

Figure 1: Architecture of Adaptive Monitoring System

Since our aim is to secure the monitoring system and information communication between the central monitoring system and the target distributed monitored system in the adaptive distributed system while keeping adaptation, the monitoring architecture of the distributed monitoring system is redesigned in such a way that the information gathered for the purpose of adaptation is checked for its security-critical before it is sent to the central monitoring system using the Event Analyst as indicated by the bolded doted rectangle (see Figure 1). This leads us to checking events that are created

in the target system whether the event is the event of interest or not. Hence, the event detected by the local monitoring system is compared against the normal events that the system creates for healthy operation. This requires having a repository of events created as a result of healthy operation in a repository (R) (see Figure 1). Hence, the event analyst component takes an input (event detected) from the trigger recognizer and data collector components of the event recorder of the local monitoring system and checks for its security-criticality using the security metrics locally.

The newly redesigned distributed monitoring architecture results in a model we call Distributed Adaptive Monitoring System (DAMS) and it has components: Event Recorder (ER) that interfaces with the target distributed system and captures the information generated as a result of the event triggered due to the changes in its environment. Besides, this ER is expected to identify the events of interest from the events created as a result of healthy operation using the other component called *Comparator* and passes the same to the Event Analyst component. The other newly added component is the *Event Analyst* (EA), which is responsible to perform some analysis on the changes detected such as on the criticality, size and detail, and support for inference of the information collected. It determines whether the information gathered from the triggered event is security critical by quantifying its attributes and compute the effectiveness of the security mechanism of the target system using the specified security metrics equation 5. Finally it sends the gathered information to the central monitoring system for processing through SSL or unsecured channel using a separate communication line called monitoring network (see Figure 2) based on the security critically of the information gathered.

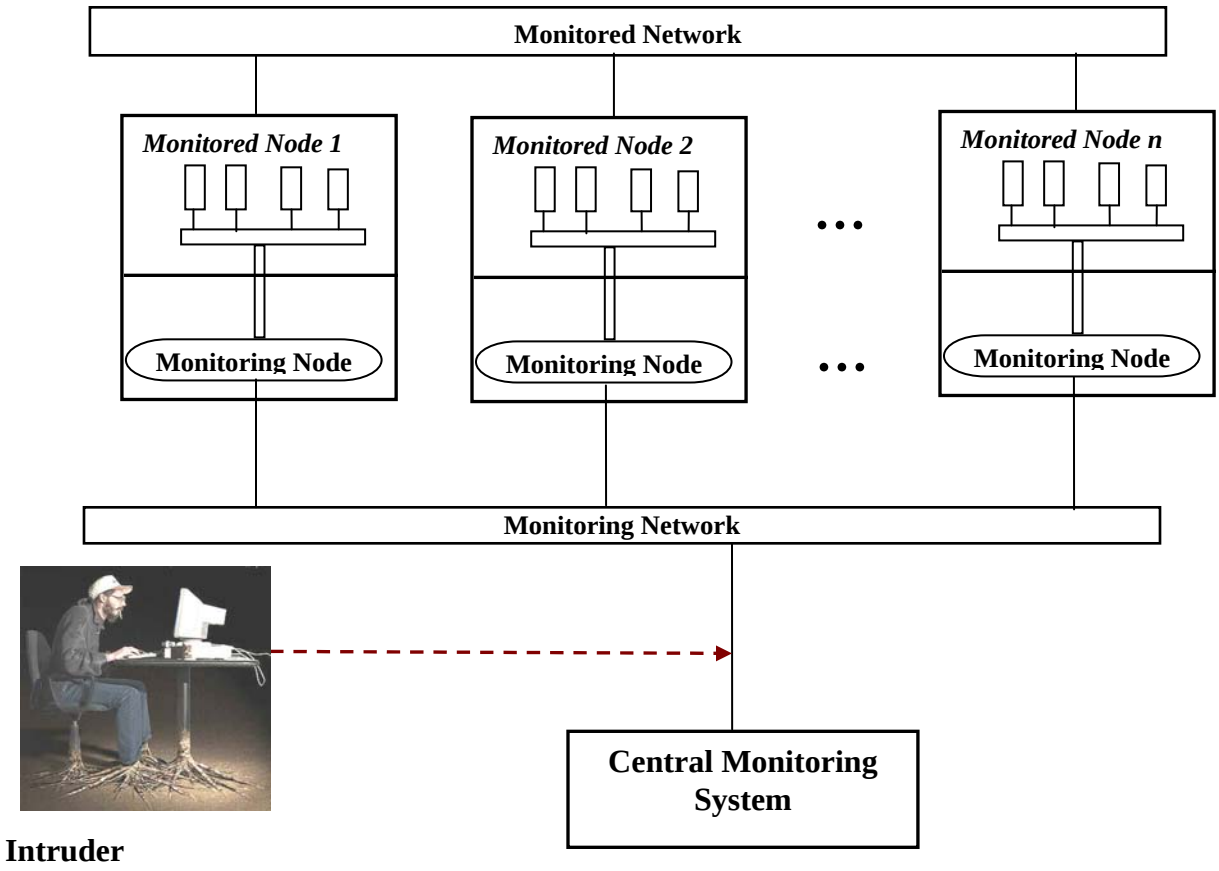

Figure 2: Distributed Monitoring Architecture [21]

Allowing the monitoring system to gather information for the purpose of adaptation may cause security problem. Since the information gathered have the probability to be intercepted by intruders as it is being sent to the monitoring system (see Figure 2), the Event Analysis (EA) component of the adaptive monitoring system must perform analysis on the information gathered and compute the metrics value using Equation 3 or 5 to determine how much risk it is if sent over the unsecured channel. Based on the analysis, DAMS facilitates the decision to be taken by the local monitoring system before it sends the gathered information to the central monitoring system. If the analysis indicates a high risk of sending the gathered information on unsecured channel, then a secured channel or an encryption will be used (see Figure 3).

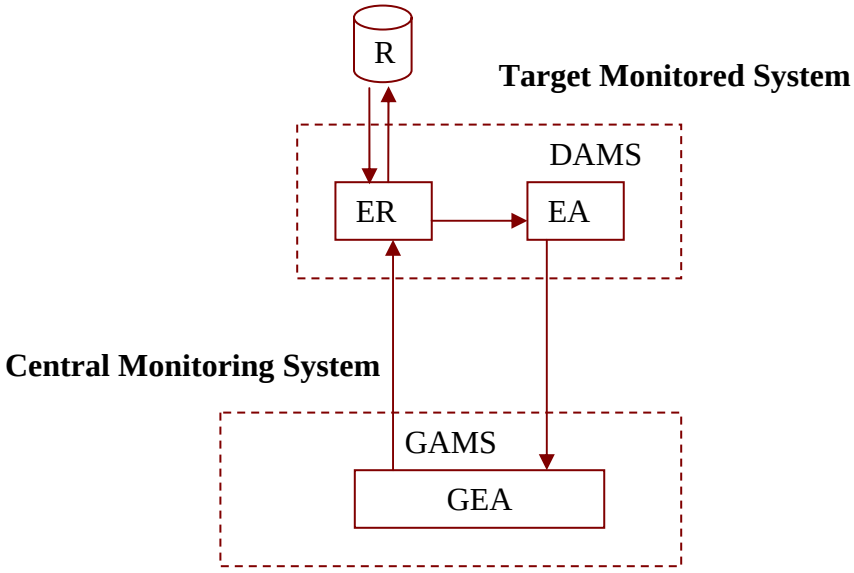

Figure 3: Adaptive Monitoring Model

Even though it has some effects (to be investigated in the future) on the performance of the target monitored system, implementing DAMS on each target system has its own advantages. One of such advantages are that it reduces the probability of the information to be intercepted by an intruder to the minimum, i.e., if the sensitivity of the information gathered from the target monitored system for the purpose of adaptation is known before hand, DAMS will automatically use a secure channel or encryption function when sending the information to the central monitoring system. The other advantage is that, if the computation of the security metrics is done on the central monitoring system, all nodes will send the detected information to it and the central monitoring system becomes busy in computing the security-critically using Equations 3 or 5 rather than performing some global activities. Hence, performing the security metrics computation locally reduces such computing burden on the central monitoring system and makes it available for other jobs.

DAMS is assumed to be integrated to the monitored system. Hence, the hardware monitoring unit is connected to the external bus of the monitored system through the bus interface module that encapsulates the monitored system's bus activity so that the event recorder is independent of the bus structure of the monitored system. The hardware unit should be directly connected to the monitored processor, by passing

the cache of the memory management unit and it is accessed as a coprocessor of the processor of the monitored system, i.e., the processor controls the coprocessor's activity.

The central monitoring system can also gather information through local monitoring systems for further adaptation. Hence, unless the proper security mechanism is in place the communication of the central monitoring system with the local monitoring system on each target is still vulnerable to security threats. Because of that, it is necessary to propose another monitoring model, which will be implemented on the central monitoring system called Global Adaptive Monitoring System (GAMS). GAMS is responsible to perform analysis on the content of the message to be sent to the distributed monitoring system. Hence GAMS has one component called Global Event Analysis (GEA) responsible for such analysis and take proper action as EA in DAM.

4. Implementation

Adaptive distributed systems are expected to listen to their execution and interaction environments in order to gather information about changes in their computing environment for the purpose of adaptation. Hence, a system listens to the activities executed at the *process level* since process is the minimum program unit that can easily be monitored and managed, and the collection of information useful for adaptation purpose from the recognized events by collecting keyword from the memory address where the triggered events are stored. After collecting the necessary information, the monitoring system quantifies the information based on predefined parameters, e.g. its criticality in the context of security. Finally, it computes the level of risk and security threat that monitoring may impose if the collected information is disclosed to unauthorized agent when it will be passed to the central monitoring system using the adaptive security metrics equations proposed in the previous sections.

To secure information during temporary storage locally at the target system and during communication with the central monitoring system, the target monitoring system determines how to pass the gathered information based on its criticality.

Information with quantified metrics values higher than a given threshold, i.e. security critical information, will be sent to the central monitoring system by using a secured channel – a costly and dedicated channel for the transmission of critical system.

Implementation of the proposed security mechanism in an adaptive distributed system requires measuring a number of parameters (*criticality*, *size, detail* and *support for inference of the attributes)* necessary for quantifying the gathered information for the purpose of adaptation. Measurement according to Pressman [23] is the process by which numbers or symbols are assigned to the attributes of entities in the real world by defining them according to some clearly defined rules. In other words, measurement helps to better understand the attributes of models we propose for the adaptive monitoring system. But unlike other engineering disciplines, software engineering is not grounded in the basic quantitative laws of physics. "Direct measures such as voltage, mass, etc, are uncommon in the software world". Hence, software measurements and metrics are often indirect and can be used to make important decisions.

As we have discussed previously, information that the monitoring system gathers for the purpose of adaptation may or may not be security critical. The criticality of the information itself may vary depending on the damage that can be inflicted if it is intercepted by a third party intruder. For this reason, computer assets, both information and hardware have definitely different degrees of criticality that may vary from time to time or over a certain period. That is, information that is critical at one time may not be critical at another time and all information may not have the same values of criticality attributes. Hence, the degree of criticality of given information at a certain time has to be quantified some how in such a way that it will help us to take proper decisions. For example, we can assign the criticality matrix table we have proposed in

Table 3 below.

Table 3: Information Asset Classification table (Example)

Information Classification	Keyword	Weighted Value	Remark
Extremely Critical	Attempt to access highly confidential information	5.00	Requires 100% security
Critical	IP Address, Network ID	3.75	Requires 85% Security
Moderate	Listing to Ports	2.50	Requires >50% security
Public	Others	0.00	Requires No security

Functionalities of an adaptive distributed monitoring system can be summarized using the use case model shown in Figure 4. A use case diagram models system functionalities at high-level of abstraction.

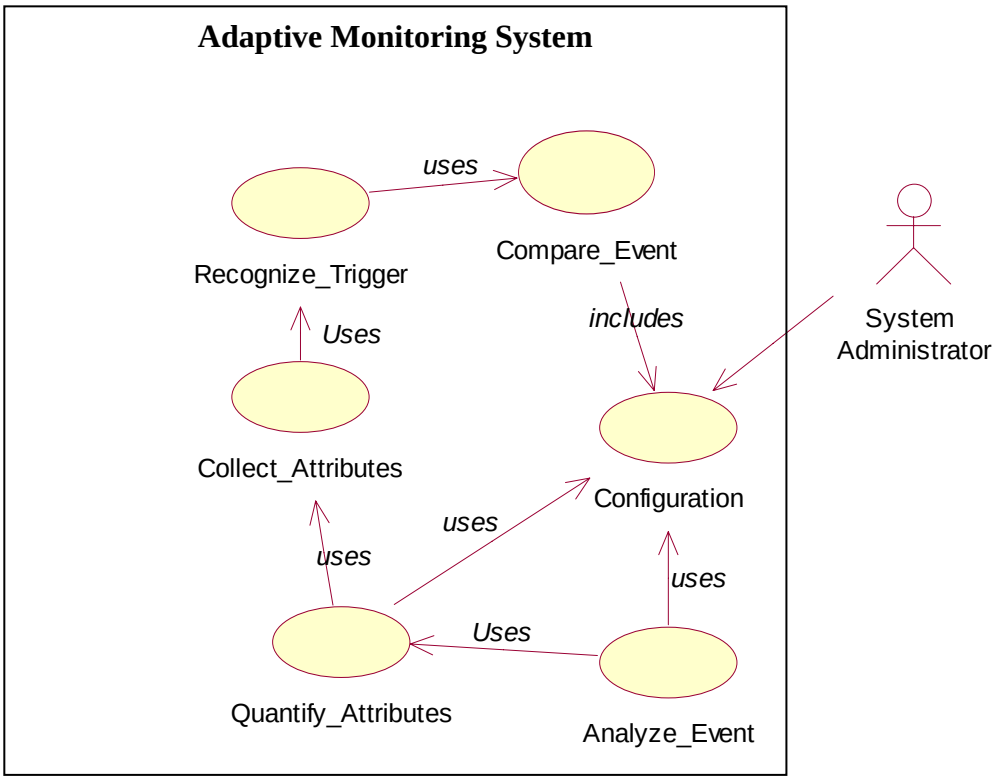

Figure 4: Use case diagram for Adaptive Distributed Monitoring system

A more detail description of the functions and functionalities of the adaptive distributed monitoring system and its components are provided below.

Event listener and trigger

One of the major activities of an adaptive monitoring system is listening to changes occurring in its environment. This may include events created by running programs, changes in the resource utilization, etc.

Monitoring of a distributed system can be done through hardware, software or a hybrid of hardware and software and the monitoring activities which are defined as the process of dynamic collection, interpretation and presentation of information concerning objects or software processes under scrutiny. Hence information can be extracted dynamically based on events and triggers as follows.

In hybrid monitoring, triggering is accomplished by inserting instructions into the target system to start recording in the hardware and the software specifies the events to be detected in the monitored target program. For this purpose it has the event recorder component described in section 4.

There are two triggering approaches:

- One has a set of predefined addresses to trigger event recording. For this reason the monitoring unit is mapped into memory address. Each address represents an event. When a predefined address is detected on the system address bus, the monitoring device records the address and the data on the system bus. This approach is known as *memory mapped monitoring*.
- The other approach uses the coprocessor instructions to trigger event recording. The recording unit acts as a coprocessor that executes the coprocessor instructions.

We call the module, which uses one of the above approaches to recognize the events that are the basic way to recognize changes in the distributed environment **Recognize_Trigger**. The outputs of this module are the keywords collected from

the process under execution and are the main data component from which we can compute effectiveness of the target monitored system.

Collect Attributes

After recognizing the events triggered and as a result of changes in the environment, the adaptive monitoring system collects the information and computes its metrics based on the preset values of the attributes such as its criticality, size, detail and support for inference. In the event listener and recognizer section, we have discussed that the module listens and collects keywords from the process execution. It is these keywords that help us to compute the values of the attributes.

To determine whether or not keywords are security critical, we have designed a security criticality matrix that shows the level of importance of our information asset. Hence, the attribute collector module called *Collect_Attributes*, checks the keyword in the security criticality table and determines the level of criticality and the weighed value assigned to it (see

Table 3).

Quantify Attributes

To compute the effectiveness of the target monitoring system, we need to quantify the attributes collected based on their security criticality. By quantification of attributes, we mean that assigning the numeric values that best describes the importance of the information even though it is not a direct measure like in other disciplines. To quantify attributes, we use a module called *Quantify_Attributes* that quantifies the identified attributes. Hence, the module takes the attributes as arguments and quantifies them, i.e. assigns numeric values to them, based on the following elements:

Criticality

In the context of Information Technology (IT) security and business continuity, the value of information assets is indicated in terms of their criticality and

sensitivity [17, 18]. Hence, criticality is the attribute used to rank the importance of information asset to its owner.

Determining the criticality level of information asset basically requires classification of the information into several categories: *confidential*, *sensitive* and *public* [18] based on the importance of the information. Therefore, the process of information asset classification enables us to know whether or not the information assets we are handling are critical so that we may handle it properly and apply a reasonable and effective security mechanism.

The quantified criticality value of the information is the user assigned value during information asset classification. Basically, if the attribute is among the information assets that are classified as critical or confidential, the effectiveness of the security mechanism must be above 85 percent. Table 2 in section 4.1 summarizes the information classification and assigned numeric values. As we have stated above, the criticality of information varies from time to time, and for this reason the proposed information classification table in section 4.1 based on the criticality of our asset varies from time to time showing the up-to-date information classification. The user has to update the table from as frequently as necessary. Hence, the table can be considered as dynamic table and used as lookup table during implementation after the event is identified.

To information classified under extremely critical category, we assign a weighed value of 5 in respect of the size, and support for inference of the information. Therefore, for whatever values of the size and support for inference, such information has to be taken care of when transferring among the nodes in the distributed system. For example, by using some encryption mechanism, we can protect the information theft. For information classified as critical, we also assign a weighed value of 3.75, and irrespective of the values of the other attributes, we have to consider it as extremely important information. For other classifications, moderate cares will be taken depending on the values of the other attributes, i.e. size and support for inference.

Size

The value of the size attribute of information gathered for the purpose of adaptation is the volume of the information to be sent to the monitoring system. Basically, the size attribute of the information gathered is the totality of the information to be transferred among the target monitoring system and the central monitoring system. The aim of the size attribute is that if the targeting monitoring system is going to transfer information in bulks, it indicates that there may be a security breaches. In addition to this, such a bulk data transfer may consume the network resource. Hence, size attribute can be computed by listening to the network and by analyzing the number of bits of information transferred.

Support for inference

The attribute *support for inference* is the possibility to derive security relevant information about other data objects from a given data security relevant information. This value theoretically requires investigating and analyzing the information to be sent to the monitoring system for the purpose of adaptation. In this thesis, we assume that each client will send to the monitoring system information about changes on itself. Hence, the support for inference of attributes collected for the purpose of adaptation is almost negligible.

Securing information in transmission

The last module in the proposed adaptive monitoring system is a module that computes the effectiveness of the security mechanism of the target monitored system and determines how to communicate information gathered for the purpose of adaptation from the target monitoring system to the central monitoring system - called the *Analyze_Event* module.

To compute the effectiveness of the target monitoring system, we can make use of the adaptive security metrics proposed in section 4.1. Based on the result of the security metrics equation, we can determine how much risky the target system is

and can determine whether or not it is necessary to use secured channel to transfer the gathered information to the central monitoring system.

Activity diagram of a Distributed Adaptive Monitoring System (DAMS) can be summarized as indicated in Figure 6.

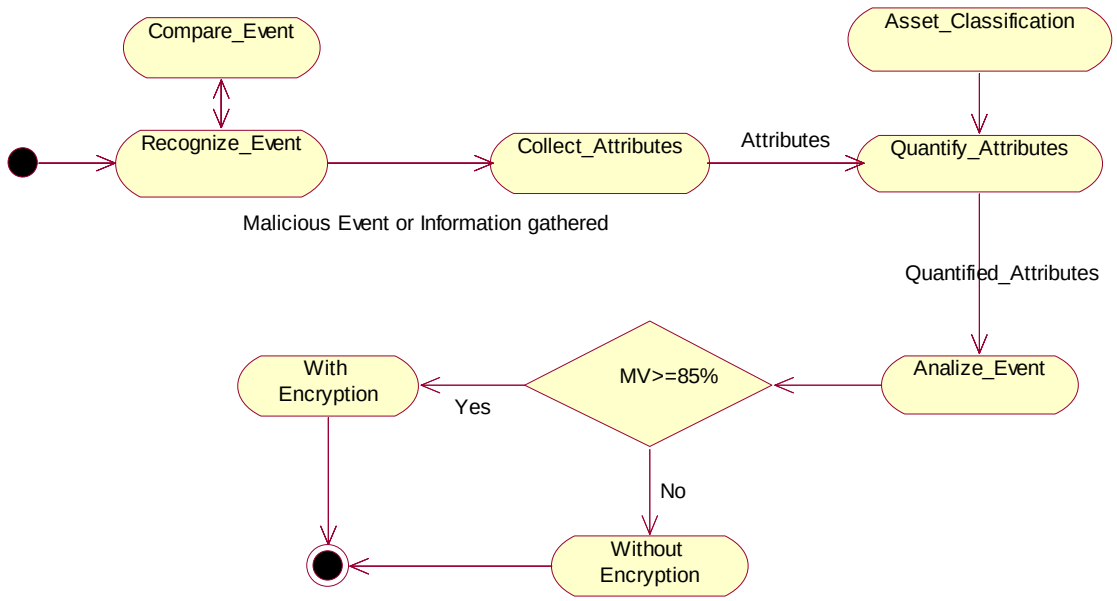

Figure 5: Activity diagram for Adaptive Monitoring System

4.1. Limitations of the Implementation

This section outlines a sketch of an architectural model of an adaptive monitoring system and the diagrams given in this section are intended to show how the proposed framework can be realized. The discussion of implementation issues is intentionally kept at higher level of abstraction and more emphasis is put on the underlying theoretical analysis. The following points can be mentioned, among others, as the main reasons that forced us to focus on the theoretical aspects:

- The fact that there is no real and feasible platform and computing environment at the department to implement the framework;
- There are no resources in the literature on the implementation issue of the adaptive distributed system as the area is relatively new; and
- It could have been possible to start the implementation from scratch, but the resources required for setting up and developing the modules that work at process level requires long time.

Hence, we believe that elicitation of the theoretical foundation underlying the assessment of the impacts of monitoring for adaptation may have on distributed system; development of a framework for a monitoring system that adapts itself to the changes in its execution environment; and a full fledged implementation of the proposed framework would have been too ambitious undertaking. Consequently, we decided to address the first two elements and third issue is briefly scratched and considered as one of the issues for future work.

5. Major Achievements, Conclusions and Future Work

In this thesis, we have discussed and assessed security threats in the Adaptive Distributed System which results as a result of adapting the distributed system. As we have discussed in the thesis, we can adapt a distributed system by allowing it to gather detailed information through its monitoring component. On one hand, allowing the monitoring component to gather detail information increases the adaptive capability of the distributed system and on the other hand, allowing the distributed system to gather detail information can cause considerable security problem if the gathered information for the purpose of adaptation is available for the man-in-the-middle by overtaking the monitoring component.

Distributed systems are vulnerable to security threats in such a way that modern distributed systems use interceptors that help the system to extract useful information and store it in a log file for further analysis to make useful decision. Hence, we argued that if this useful information is available for intruders, then the intruders can act as

one of the normal node in a distributed system so that it can intercept any information communicated in the distributed system. For this reason, in this thesis we have made assessment and found the following major achievements.

5.1. Major Achievements

One of the major sources for intruder according to our findings is the monitoring component of the adaptive distributed system. Monitoring component is the means to gather detail information for distributed systems to adapt themselves based on the changes in the environment. Basically it is this information that needs security. Therefore, if we can secure information gathered by the monitoring system we can achieve on the required secured system. Hence, to have a secured adaptive distributed system, we have proposed an adaptive monitoring system.

Security issues in distributed information systems, whether adaptive or not, are serious concerns. Among the many types of threats, those occurring during communication and those in the form of unauthorized attempts to access stored information are the main ones. Security issues in distributed systems are classical problems, which have partly been solved using techniques such as cryptographic systems, access control and auditing mechanisms. We argued that there are situations where these techniques can't be applied. For example, there are countries that don't allow encrypted packets to pass over their network. Hence, to protect the monitoring system and the detailed information gathered for the purpose of adaptation from the two possible threats, we have proposed an adaptive distributed monitoring system that gathers information from the targeted monitored system and sends to the central monitoring system. To avoid and/or reduce threats that occur during communication, the proposed adaptive distributed system sends the information to the central monitoring system through the secured channel based on the criticality of the information or restrict itself not to gather detailed information. If the information is not security critical, it sends the information to the central monitoring system using

the unsecured communication. In doing so, we believe that an adaptive distributed monitoring system better understands its environment and takes proper action before critical information is intercepted by the man in the middle.

To know whether the information gathered is security critical or not, we have proposed a security metrics that measures the effectiveness of the target security mechanism through the security metrics equation. We believe that the importance of a given information varies from time to time. In other words, all information is not critical. Hence, we have designed a security metrics that measures the effectiveness of the security mechanism based on the classification of our information system asset and the assigned weighted value. Our intension is that, users can change this security metrices table (the weighed value) from time to time based on the importance of the information and hence adaptive.

We have also discussed that one can't have hundred percent security mechanisms for the system and very small security mechanism means that there is no security for the system. In other words, if we need a hundred percent security for our system, definitely we compromise the required quality of services expected from the system. Hence, we have considered different scenarios and come up with the conclusion that there must be a level to what extent we can allow the distributed monitoring system to gather information for the purpose of adaptation. Hence, we have proposed a dynamic level, i.e., configurable level based on the need of the user as Excellent (above 85% security mechanism), Above Average (71 to 85 percent), Average (51 to 70), Below Average (26 to 50), and Poor (below 26 percent security mechanism).

5.2. Future work

As we have discussed in the major achievements, to protect the adaptive distributed system from the two potential threats, we have proposed an adaptive distributed system that measures and takes proper decision. Even though we have proposed such a system to protect our information asset, the *impact* of the proposed solution on the quality of service we expected from the system must be investigated. Hence, in the

future, we will investigate the impact an adaptive distributed monitoring system has on the adaptive distributed system and refine the adaptive security metrics.

The monitoring system gathers information from the event triggered as a result of malicious act or information gathered for the purpose of adaptation. Hence, the collection is done when the instructions are going to be executed. Therefore, can't we apply this work to identify all malicious activities such as virus, malwares, etc, and take proper actions before they damage our system? Can we apply the distributed adaptive monitoring system in the area of real time environments? Is the quality of service required from the adaptive distributed system met? These are some of the question that are to be investigated in future work.

REFERENCES

[1] Francisco José da Silva e Silva, Markus Endler, and Fabio Kon: Dynamic Adaptation of Distributed systems, 16th European Conference on Object-Oriented Programming, 2002.

[2] Partha Pal, Rick Schantz, and Joseph Loyall: Timeliness in Auto-Adaptive Distributed System, BBN technologies, Cambridge, USA, 2004.

[3] Matti A. Hiltunen and Richard D. Schlichting: Adaptive Distributed and Fault-tolerant Systems. International Journal of Computer Systems Science and Engineering, 11(5):125-133, June9, 1995.

[4] Wen-ke Chen, Matti A. Hiltunen & Richard D. Schlichting: Constructing Adaptive Software in Distributed systems, in the Proc. of the 21st International Conference on Distributed Computing System, (ICDCS-21), pp. 635-643, Mesa, AZ, 2001.

[5] Ilwoo Chang, Matti A. Hiltunen, and Richard D. Schlichting: Affordable Fault Tolerance through Adaptation, Parallel and Distributed Processing, LNCS 1388, pp. 585-603, April 1998.

[6] Demissie Aredo and Sule Yildirim: Security Issues in Adaptive Distributed Systems, in the Proc. of the 14th European Conference on Information Systems (ECIS2006), June 12-14, 2006, Gøteborg, Sweden.

[7] J. Voas, A. Ghosh, G. McGraw, F. Charron and K. Miller: Defining an Adaptive Software Security Metric from Dynamic Software Failure-Tolerance Measure, in the Proc. of the 11th Annual Conference on Computer Assurance (COMPASS'96), Gaithersburg, Maryland, 1996.

[8] Hanno Langweg and Einar Snekkenes: A Classification of Malicious Software Attacks, Norwegian Information Security Laboratory-NlSLab, Department of Computer Science and Media Technology, GjØvik university college, Norway, 2004.

[9] Shyam R. Chidamber and Chris F. Kemerer: A Metrics Suite for Object Oriented Design, IEE Transaction on Software Engineering, Vol.20, No.6, June 1994.

[10] Stefano Salsano, Luco Veltri, and Donald Papalilo: SIP Security Issues: The SIP Authentication procedure and its processing load, IEEE Networks special issue on Network Security, Nov/Dec 2002.

[11] Valentina Casola, Massimiliano Rak, Antonino Mazzeo, and Nicola Mazzoccca: Security Design and Evaluation in a VoIP Secure infrastructure: A policy Based Approach, IEEE Proceedings of International Conference on Information Technology: Coding and Computing (ITCC'05), computer society, Las Vegas, April 2005.

[12] Marianne Swanson, Nadya Barto, John Sabato, Joan Hash, and Laurie Graffo: Security Metrics Guide for information Technology Systems, NIST Special Publication, pp 800-55, July 2003.

[13] IEEE 802.5 standard, 1998.

[14] Foundstone Strategic security: Information Security Metrics, Using Foundstone's FoundScoreTM to assign Metrics and Measure Enterprise Risk, April 2003.

[15] The Common Object Request Broker: Architecture and Specification, Version 2.3, 1999, Object management Group Inc.

[16] Shahid Nazir Bhatti: Why Quality? ISO 9126 Software Quality Metrics (Functionality) support by UML Suit, ACM SIGSOFT Software Engineering Note 30(2), Vol. 30, No. 2, March 2005.

[17] APS 131: Inventory and Classification of Information assets, Draft Vol.8 Administrative Policy statement, University of Colorado, (8/11/05).

[18] http://www.sinclair.edu/about/information/usepolicy/pub/infscply/Identification_and_Assessment_of_Assets_and_Risks.htm, Identification and Assessment of Assets and Risks, visited on 24/12/2006.

[19] Dieter Haban and Dieter Wybranietz: A Hybrid Monitor for Behavior and Performance Analysis of Distributed System, IEEE 1990.

[20] Netscape Communications Corporation: The SSL Protocol Version 3, March 1996.

[21] Jeffrey.J.P. Tsai, Yaodong Bi, Steve J.H. Yang and Ross A.W. Smith: Distributed Real-Time Systems: Monitoring, Visualization, Debugging, and Analysis, John Wiley & Sons, 1996.

[22] Wikipedia: the free encyclopedia, Adaptation, visited on 02/02/2007, http://en.wikipedia.org/wiki/Adaptation.

[23] Roger S. Pressman: Software Engineering: A Practitioner's Approach, Six Edition, McGraw-Hill International Edition.

Declaration

I, the undersigned, declare that this thesis is my original work and has not been presented for a degree in any other university, and that all source of materials used for the thesis have been duly acknowledged.

Declared by:

Name: _____

Signature: _____

Date: _____

Confirmed by advisor (s):

Main Advisor

Name: _____

Signature: _____

Date: _____

Co-advisor

Name: _____

Signature: _____

Date: _____