Modelling an Intelligent Network Security Systems using Multi-Agents Systems Engineering

Gustavo A. Santana Torrellas
Instituto Mexicano del Petróleo
Mantenimiento y Perforación de Pozos
Eje Central Lázaro Cárdenas N°152
CP 07730, México, D.F.
e-mail: gasantan@imp.mx

Abstract - *Recent developments have made it possible to interoperate complex business applications at much lower costs. Application interoperation, along with business process reengineering can result in significant savings by eliminating work created by disconnected business processes due to isolated business applications. However, we believe much greater productivity benefits can be achieved by facilitating timely decision-making, utilizing information from multiple enterprise perspectives. To stay competitive in this current scenario, it is crucial for organizations to react quickly to changing security factors, such as virus attack, active intrusion, new technologies, and cost of disaster recovery. Such information security changes often encourage the creation of new security schemas or security improvements. Accommodating frequent systems information changes requires a network security system be more flexible than currently prevalent systems. Consequently, there has recently been an increasing interest in flexible network security and disaster recovery systems.*

Keywords: Network security, Multi.Agent Systems Security.

1 Introduction

There are two important aspects to a flexible network security system: hardware infrastructure and the corresponding planning and control software. The latter is the heart of a flexible network security system; appropriate software architecture can improve system performance significantly. In this paper, we focus mainly on the software aspect. Distributed Artificial Intelligence covers the intersection of Artificial Intelligence and Distributing Computing. Multi-Agent Systems (MAS) are commonly used in solving difficult problems in the areas of Distributed Artificial Intelligence. This paper takes the multi-agent approach in which a team of agents, each with only limited local knowledge and local information, collaborates to satisfy both local and global network security objectives. The overall behavior of the system emerges through the dynamic interactions of the agent's local behaviors.

In recent research activities author has addressed the use of MAS for the control of network security systems. Santana et al. [2] consider methodological issues for designing a flexible multi agent-based network security for authentication and authorization in mobile environments; the agents use a contract-net protocol for negotiation and dynamic security authentication in inter-domain networks. In other approach Santana, Sheremetov and Contreras [3] proposes a rather original approach to security assessment task planning for multiple policies by associating a Component Agent Platform, where a set of agents collaborate to assembly an authentication and authorization schema considering different policies to be assembled. Recently, we are working in the development of a multi-agent approach for Continuous Security Management, in which each agent controls only part of Security Information System and the whole security of the systems is configured by a mobile agent systems. Security policies are critical for network security management; the computational complexity of finding a coherent security schema for multi-domain systems grows exponentially with the total number of nodes, host, and segments of the network system. As a result, most practical approaches utilize only heuristic solutions to make the problem tractable.

2 Problem Statement and Definitions

We are developing an agent-based planning and control system for a flexible network security system with multiple policies agents. The input of the system is the general policy model of the network to be protected. The output of the system is the final security's state network. The general flow diagram that indicates the global operation of the system is shown in Figure 1. This flow is mainly divided in two stages: an off-line stage and an on-line stageThe off-line stage performs network security task decomposition. It produces a preliminary security plan that consists of a sequence of security procedures and operations and the precedence relationships among them. The input to this off-line stage is the general policy model of a system that is composed of parts. It then generates a preliminary security assessment plan based on risk analysis and vulnerabilities evaluations about accessibility and network stability. The security assessment operations that make up a preliminary information security plan are task level operations. Currently we have implemented two such operations:

- Security Requirements Assessment

- Threat & Risk Assessment

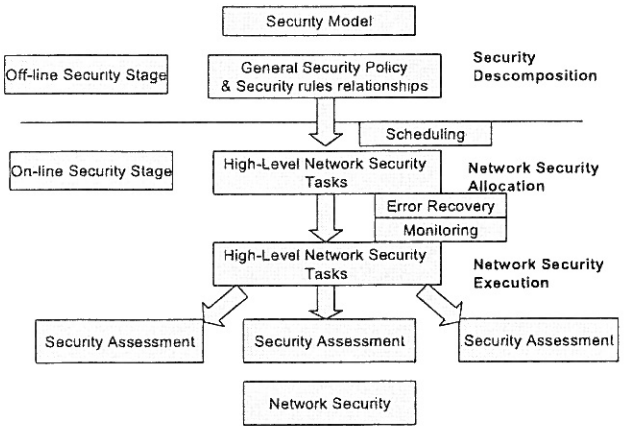

Figure 1: Global Security Flow

The preliminary information security plan is used as the input to the security assessment on-line stage. The security assessment on-line stage focuses on network security allocation and network security execution. In this stage, agents of a team collaborate and cooperate to achieve common security goals. The network security allocation phase assigns security operations to appropriate resources (security assessment agents). The task level operations are also mapped into low-level operations that security assessment agents can understand and execute. In the agent system, several special agents are added during network security execution: a collision avoidance agent (for collision-free security planning), a fault tolerance agent (for fault recovery), and a monitoring agent (for interleaving planning and execution). This paper focuses on the security assessment on-line stage. We present an agent based planning and control architecture for network security control and coordination. The multi-agent paradigm has been adopted as it provides the following advantages:

- **Homogenous Framework**: All system components are considered and modeled in a uniform and homogenous manner
- **Modularity and Scalability**: Different modules are developed and implemented independently. This approach can simplify agent component development/maintenance and make addition/removal of agents much easier.
- **Dynamic Reconfigurability**: The capability to quickly reconfigure a system enables timely response to changing business process reengineering conditions.
- **Distributed Control**: Each agent has local control with the capability to coordinate its activities through messages-passing communication. Agents can reside on different computers and security assessment controllers.
- **Fault Tolerance**: Techniques for fault detection, failure recovery, and execution monitoring during network security execution is provided to make system performance more robust.

3 Agents types

The multi-agent paradigm is based on the premise that complex problems can be partitioned into simpler sub-problems that have limited mutual dependencies. The fewer dependencies the system has, the more successful the agent-based approach can be. Therefore, problem decomposition and design of individual agent become the most important software design decisions for agent-based system. We have adopted a hybrid decomposition approach [14]. Our agent architecture is shown in Figure 2.

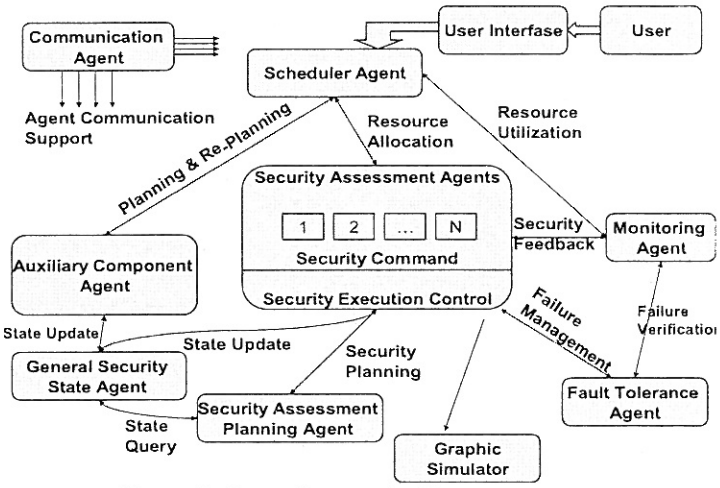

Figure 2: Agent System Architecture

We have designed the following agents in our system:

- Scheduler Agent is responsible for scheduling the detailed network security operations and assigning them to available resources (security assessment agents).
- Security Assessment Agents are responsible for the lowlevel control of individual security assessment agents, and for converting task level network security operations to low-level security requirements.
- Auxiliary Component Agents are responsible for the control of the auxiliary components, i.e. CAP – Component Agent Platform in our system.
- General Security State Agent contains both static and dynamic information of all components in the system. Static information includes connections, transactions, and dynamic information of all system components. Dynamic information includes current component configuration and any dynamic state information.

- Security Assessment Planning Agent is responsible for generating the collision-free security task of the three security assessment agents on the fly.
- Fault Tolerance Agent manages and coordinates the failure recovery process.
- Communication Agent handles message passing and message dispatching in the agent system
- Monitoring Agent monitors task planning and execution to improve resource utilization. It notifies the fault tolerance agent in case of system failures.
- Graphic Simulation Agent provides visual verification of task execution.

Typical agent architecture in our system is illustrated in Figure 5. Agents are involved in two types of activities: internal activities that lead the agent to achieve its own goals, and external activities through which the agent communicates and collaborates with others agents to achieve common goals. To support both internal and external activities, a typical agent contains the following modules:

- **Communication Module.-** The communication module is responsible for managing and controlling communications among agents. Both direct peer-to-peer communication and communication through network architecture is supported. The communication module itself consists of a low-level network interface and a higher-level communication protocol. The communication module manages two types of messages: state messages and commands messages. State messages contain dynamic state information of system components, while command messages contain other information, such as connection information, security transactions and dynamic performance, or execution commands. Command communications are implemented by priority messages, buffers and without time restrictions. A general communications protocols architecture manages the command communications.
- **Security Module.-** The security module provides an agent with an information source about the agent itself and the environment (others agents). This module has a knowledge base that stores necessary data and knowledge for the agent to perform its activities. The knowledge base includes two kinds of knowledge:
    1. **Local security knowledge base**: It contains information related to agent capabilities and its own state information.
    2. **Global security knowledge base**: It contains information of other agents, with which the agent will work. The agent collaborates and cooperates with these agents during the security assessments security assessment and network security execution stage.

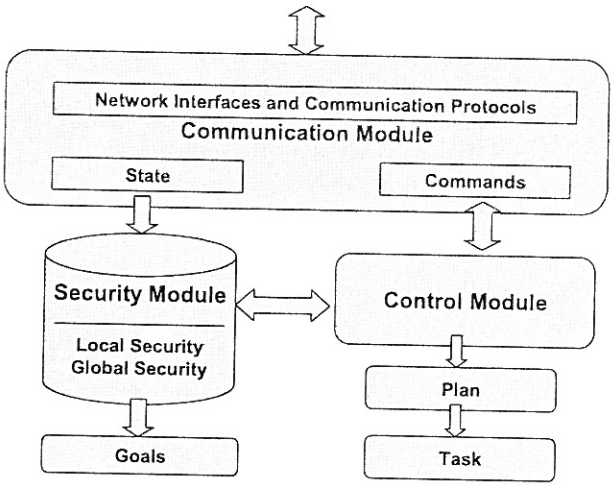

Figure 3: Agent Communication Model

- **Control Module.-** The control module determines the agent's behaviors. It operates in a goal-driven fashion. When an agent is assigned a set of tasks, every task in the set is converted into a set of goals. The agent can achieve the goals as long as its capabilities can meet the requirements. Agent goals are mapped into a collection of plan scripts, which can be achieved by the control module. The scripts are coded with the necessary parameters and low level security requirements. The control module can run either independently or cooperatively with others agents.

4 Communications among Agents

The ability to represent, query, and manipulate knowledge is crucial to agent-based systems. Finin et al [5] propose KQML, a novel language for such purpose. There are two types of communications in our system: commands and state communications.

4.1 Command communications (non-periodic communication)

A internetworking architecture has been implemented inside the communication agent to manage the command communications. The message is handled according to its time stamp. A high priority command message will be sent before a lower priority message. If two messages have the same priority, the one with earlier time stamp will be sent first. Agents communicate with each other by posting command messages on the internetworking. The communication agent is responsible for handling message dispatching sending the message to its recipient and sending the reply back to its sender. Figure 4 shows how this kind of communications is achieved. The messages that fall into the category of command communications are:

- **Control commands**: Commands related to the control of the system components (security assessment agents, security access control, and security authentication part).
- **Information request commands:** Query commands regarding the agent's state and knowledge.
- **Failure commands:** Failure report

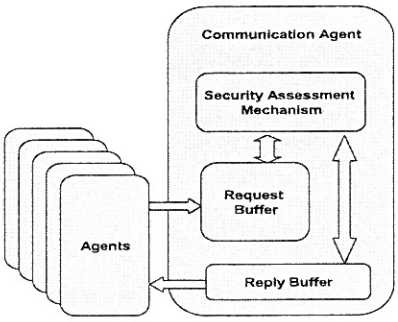

Figure 4: Command Message Managed by the Communication Agent

4.2 State communications between General Security state agent and security assessment agent (periodic communications)

During the task execution stage, security assessment agents need up-to-date state information of other security assessment agents to plan collision-free security states. This information must be updated at each time instant to reflect the current states of the security assessment agents. This state information is used by the Security Assessment planning agent during task execution. As we expect, there is intensive message passing between the General Security state agent and security assessment agents, particularly during the Security Assessment planning stage. Security assessment agents reply by sending their current state information to General Security state agent. The General Security state agent then updates all state information in it. Whenever an agent requests the state information of some security assessment, it requests the information from the General Security state agent instead of communicating directly with the security assessment agent. This approach reduces communication overhead and simplifies state information management.

4.3 State communications between General Security state agent and others agents (non-periodic communication)

Whenever a non-security assessment agent changes its state, it sends its current state to the General Security state agent. This is done in a non-periodic fashion to reduce unnecessary state polling form the General Security state agent. Figure 5 shows how the General Security state agent manages the state communications.

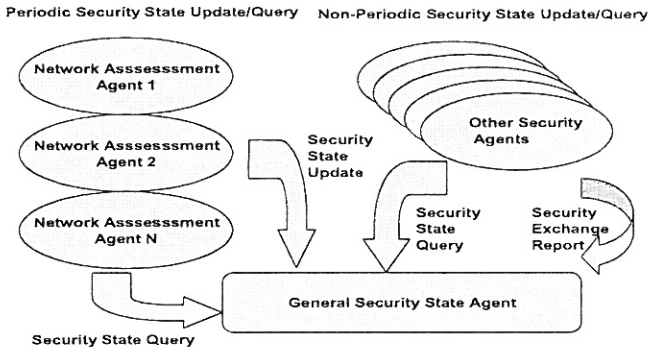

Figure 5: State Messages and General Security State Agent

5 Conclusions

In this paper, we present a distributed planning and control architecture for autonomous security assessment systems using a multi-agent paradigm. We have focused on the security assessment on-line stage (security assessment and network security execution) of network security operations. The network security execution part is actually developed, implemented, and tested in the Mexican Oil Institute. All agents except the fault tolerance agent have been designed and are currently being implemented. We belief that our multi-agent architecture is an adequate framework for flexible network security management. The system is modeled by a team of autonomous agents that develop cooperative strategies to achieve the common goals. Agents communicate with designated communication format and internetworking protocols to exchange information. We anticipated that our approach can be applied and extended to other security systems; moreover we belief in a huge field of application ranging form Continuous Security Management to e-business applications. We are currently applying and extending the planning and control strategy to a Continuous Security Management system with four security assessment agents: Security Requirements Assessment, Threat & Risk Assessment, Network Security Assessment, and Internet Security Assessment.

6 References

[Auramaki et al. 88] Auramaki, E., Lehtinen, E. and Lyytinen, K. A speech-act-based office modeling approach, *ACM Transactions on Office Information Systems*, Vol. 6, No. 2, april 1988, 126-152.

Rest of literature will be provided upon request.