8-15-1997

Intelligent Security System:Using Multi-Agent to Improve Internet Security

Lin Zeng
City University of Hong Kong, iszeng@cityu.edu.hk

Huaiqing Wang
City University of Hong Kong, iswang@cityu.edu.hk

Mathew K. Olee
City University of Hong Kong, ismatlee@cityu.edu.hk

Intelligent Security System:
Using Multi-Agent to Improve Internet Security

Lin Zeng [a], Huaiqing Wang [b], Matthew K O Lee [c]
Department of IS
City University of Hong Kong, Tat Chee Avenue, Kowloon, Hong Kong
[a] iszeng@cityu.edu.hk [b] iswang@cityu.edu.hk [c] ismatlee@cityu.edu.hk

Abstract: With the recent popularity of the Internet and the attendant proliferation of companies using it for all kinds of communication, a serious problem of security has arisen. In this paper, at first, current Internet security issues are addressed, and several existing technologies are discussed. Then we present an artificial intelligent solution which represents the current work of our Internet security project. Architecture, functions and features of the solution are described consequently.

Introduction

Today, Internet/Intranet is getting more and more attention and importance. World-wide organisations have increasingly come to use Internet/Intranet for their management and transactions with their trading partners. Along with its popularity, however, the Internet has become a common target to attack. Reports of incidents, such as attempted and successful intrusions, have grown dramatically. Several studies have shown that many individuals and companies are abstaining from joining the Internet simply because of security concerns.

Threats to the Internet are often mentioned as system intrusions, network break-in and commercial espionage, etc. Recently, password sniffing and sequence number attacks were successfully launched, too. Even more, there are reports of distributed and co-ordinated attacks[1]. Hacking software are widely available on Internet, some of which are automatic and intelligent. Threats don't just come from outsiders. Instances of internal interference are growing. Evidence shows that it is the internal and not the external threats that must concern us the most.

Nobody can ignore firewalls [2] when talking about Internet security. A firewall is an intermediate system that can be plugged between a trusted network and the insecure Internet, providing a static traffic routing service either at the network layer using screening router, or at the application level using proxy servers or application-layer gateways. Though very effective when dealing with some classes of attacks, firewalls fail in many cases. For router-based firewalls, packet-filtering rules are hard to define, since information in the TCP/IP package alone is inadequate to provide the level of resolution often needed. Proxy servers and application-layer gateways must be built for every single application. This inflexibility leads to difficulty as new services are added. A firewall alone is far from enough.

Meanwhile, with well established encryption and authentication technologies, user identification and data confidentiality are available for *point to point* communication on Internet. Consequently, KDS (Key Distribution Systems) systems, which are to establish and store encryption keys, and distribute them to network users, have been developed in succession, e.g. MIT's Kerberos and IBM's KryptoKnight. The existing KDS systems, however, have weak points by its nature. For example, Kerberos relies heavily on static and one-time check of user's ID and password, thus providing a so-called 'all or nothing' mechanism which is somehow easy to be penetrated. A real secure KDS system is still on its way to evolve.

As mentioned above, although effective in some directions, the current Internet security solutions are far from satisfactory. Threats are all-around and hard to predict, while the methods we are using are always fragmented and restricted on one point; Attackers are persistent and flexible, while the defence is static, fixed and passive. What we need is a comprehensive and automated [3] protecting mechanism.

Multi-Agent based Internet Security System

The notion of Multiple Intelligent Software Agents was proposed to address this challenge. Like other intelligent agent systems we had built [4][5], agent enjoys the following properties: (1) autonomy, (2) social ability (agents communicate with other agents) (3) reactivity, and (4) proactivity. Deployed in a distributed environment, multi-agent systems can comparmentalize specialized task knowledge, organize themselves to avoid processing bottlenecks, and can be built expressly to deal with dynamic changes in agents and information-source. We believe that a distributed artificial intelligent approach is superior for issues of Internet security.

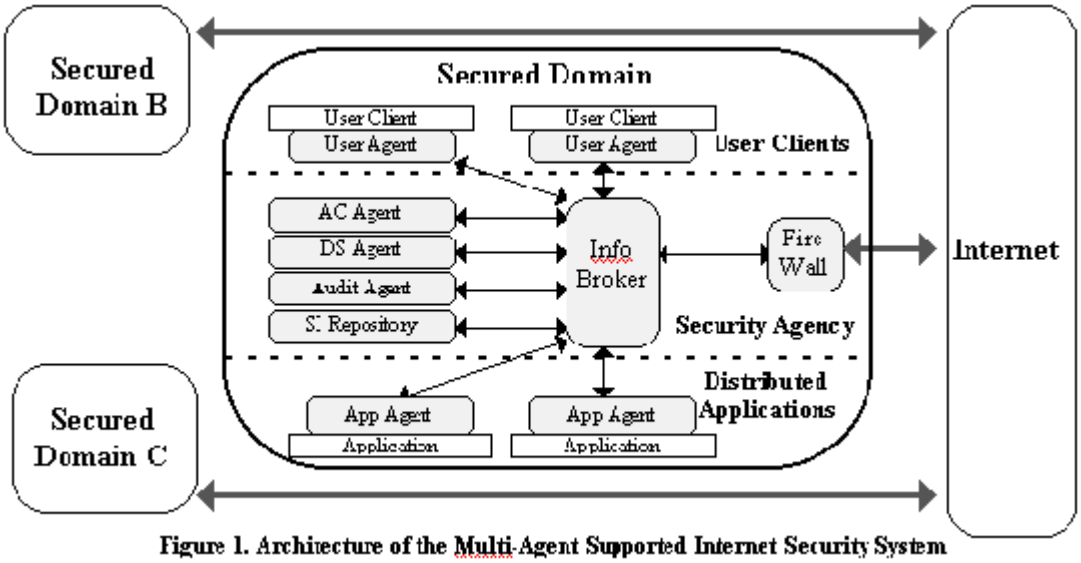

Figure 1. Architecture of the Multi-Agent Supported Internet Security System

As shown in Figure 1, we present a multi-agent security system which is focusing on one network domain (subnet) only. Five kinds of agents are deployed in the system. Following is a brief description of the components respectively.

User agent The function of *user agent* is two folds. On one hand, it is an security interface between users and the secured network resources. It provides functions of authentication, encryption and integration checking, which safeguards *point to point* communication between users and network applications. On the other hand, being an expert agent, the *user agent* can detect network violations. By monitoring and analysing a user's activities, and comparing them with historical profile of the user, it can determine whether user's current behaviour is acceptable. It also co-operates with other *user agents* invoked by the same user running in the secured network (for example, the user may login to the network through more than one site) to make the judgement.

Application agent (App Agent) *Application agent* is acting as an interface between one distributed application and the secured network. Like *user agent*, it also provides encryption and authentication. Thus, by the joint work of *user agent* and *application agent*, a reliable and confidential *point to point* communication is established. The expert component of the *application agent* is a rule-based system, which makes attack detection based on stored rules regarding security information specific to the application, such as attack scenarios, application vulnerabilities and expected system behaviour.

Audit agent The *audit agent* is an expert agent with a rule-based decision mechanism. It co-ordinates with *user agents* and *application agents* to detect the intrusions related to both of them. The decision is also made upon information exchanged from other *audit agents* of peer network domains.

Access control agent (AC Agent) Unlike most of the security systems that are user-based, the access control function of this system provides role-based control mechanism. It determines the access of each user by the classification of the user's role, but not the relationship between the user and the network applications

which the user accesses. It issues timing ticket for accessing according to different roles of users and different feature of applications. This functions is also supported by intelligent processing ability.

Domain Security Agent (DS Agent) There is only one *Domain Security Agent* for each network domain, which acting as the security center. It contains the overall knowledge of the network domain and has three functions. First, it acts as the central encryption key distributor, providing and allocating the keys for *point to point* communication between different agents. Second, when a intrusion is detected and reported, it can automatically produce plans for countermeasure and recovery. Finally, it has the functions for system administration.

Other Components A *Security Information Repository* is used to store security information for the network domain, including access information, user profiles, application profiles, encryption and authentication keys, etc. *Information Broker* runs as a co-ordinator through which agents communicate to each other in the networks. It is not only for the security agents but also for all of the accessible applications in the subnet. A *firewall* is used as the interface between the subnet and the Internet. Its main function is to filter out the requests which intend to bypass the *Information Broker*. Additionally, in order to handle the requests from users in other network domains, agents may contact through Internet with their peer agents in remote networks.

Features of the System

Integration This system organically integrates many currently available security technologies, such as authentication, encryption, firewall, and role-based access control, so that a comprehensive and overall solution can be achieved.

Intelligence Unlike normal prevention-based approach, this multiple intelligent agent system can actively protect networks from threats. By its intelligent processing abilities, it is able to analyse risk predictively, monitor activities dynamically, detect intrusions automatically, and plan and operate countermeasures intelligently. For example, *user agents* and *application agents* keep on monitoring the network activities and make judgements of violation. *Audit agent* seeks information from *user agents* and *application agents*, makes decision by its stored rules, and submits the result to *Domain Security Agent* when an intrusion is detected. *Domain Security Agent* schedules countermeasure and recovery actions upon its knowledge base, and executes the plan. Actually, the aggregate intelligence of the agents makes up to a security agency that fulfils the security goal of the whole subnet domain.

Platform Independence In an open network as Internet, users may access the secured network domain from heterogeneous platforms. This may make difficulties to build a widely accepted *user agent*. In this system, we solve the problem by using Java as coding language (just for user agent) so that when user login to the secured network, *user agent* can be automatically downloaded and run in a pre-installed Java virtual machine environment. This solution leads to platform-independence which is difficult for traditional methods.

Reusability and Scalability For every distributed application on the secured network, there should be an *application agent* built just for it. While more applications are added, huge work would be involved in building the corresponding *application agents*. This traditional problem for distributed systems is solved in this system by separating the control mechanism from security information specific to the application. The pieces of agent code for control logic can be copied from one agent to another. When building the agent, the only thing should be done is to coustomize the security profile (security rules and conditions) specific to the application. This reusability feature would be more significant while the network is expanding and applications on network are increasing.

Conclusion

This system integrates many available technologies to provide a comprehensive protection. Furthermore, it is not only a simple integration, but one with intelligent processing abilities. Comparing with the traditional static and passive architectures, this system can handle the threats pre-actively and intelligently. We believe that this artificial intelligent approach will contribute to the improvement of Internet security, and will ultimately form the security basis for a new generation of Internet/Intranet systems.

Main References

1. Frederick B. Cohen: *A Note on Distributed Coordinated Attacks*, Computers & Security, 15 (1996), pp. 103-121
2. Rolf Oppliger: *Firewalls aren't Enough: Authentication and Key Distribution Systems*, Computer Security Journal, Volume XI, Number 2,1995, pp. 15-24
3. Mukherjee, et al: *Network intrusion Detection*, IEEE Network, May/June (1994) pp. 26-41
4. Huaiqing Wang, et al: *APACS: a Multi-Agent System with Repository Support* , Knowledge-Based Systems, 9 (1996) pp. 329-337
5. Huaiqing Wang: *Repositories for co-operative Information Systems*, Information and Software Technology 38 (1996) pp. 333-341