Study on Service-Oriented Security Architecture

Wenjun Cheng
Air Force Command College
Beijing, China, 100097

Xiaosu Zhan
Academy of Military Science
Beijing, China, 100091

Shaohua Zhang
Air Force Command College
Beijing, China, 100097

Abstract—**Security has become the key issue in the development of large-scale information system. Piling security products and security technologies simply rather than functional integration and making full use of these products and technologies will cause resource wasting, and not meet the needs for flexible and diverse security requirements. This paper proposes a security architecture design based on SOA. Firstly, security devices and technologies are decomposed into basic security components which form the basic security service layer. Secondly, various extended security components that make up the extended security service layer are realized based on functional combination and process control. Then security services are provided through these two levels for upper security applications. At last, this paper gives the core architecture design of the security service core for dealing with the scalability bottlenecks in distributed system. The design solves effectively the security of large-scale information system.**

Keywords- architecture; Service Oriented Architecture (SOA); network security; cache

I. INTRODUCTION

The large-scale information system has a complex structure deployed with extensive applications and distributed functions [1]. In order to achieve the security of large-scale information system, various types of security technologies and security products are applied to this kind of system including VPN, firewall, instruction detection systems, anti-virus gateway and vulnerability scanning. This approach can provide security services from different technical levels and monitoring levels, but the services can not been effectively integrated and the upper security application could not take an efficient use of them. Thus this paper proposes a kind of service-oriented security architecture and the detail design of interconnect core.

II. KEY PROBLEMS TO BE ADDRESSED

The security architecture design of the large-scale information system needs to solve four key problems: complexity, consistency, variability and invisibility. The best way that simplifies the complexity and reduces the variability is to stratification the system into levels. According to the theory of stratification, we can find the meeting point of security services and security applications, and construct the security infrastructure platform of large-scale information system. The security infrastructure platform decomposes various security services that provided by those security products and technologies, and converts these security services to standard SOA services for the upper security applications. At the same time, the upper security applications can integrate, manage and apply various standard services based on security policies and business requirements. Therefore, the details of security products and technologies can be shielded and the problems such as complexity and variability can also be solved.

III. DESIGN OF SERVICE-ORIENTED SECURITY ARCHITECTURE

The security infrastructure platform of large-scale information system can refer to the idea of service-oriented architecture. SOA is a kind of software architecture which is designed to satisfy soft requirements by using services [2, 3, 4]. The network nodes distribute their own resources with independent services to other participants of the same SOA environment, and participants can use these resources by Web Services or other technologies based on services. Each SOA is made up of various services which are loosely coupled and interoperable. The interfaces which define the services maintain the relation between users and providers, and hide the concrete implementation of the services.

According to the theory of stratification and the design idea of SOA [5], the security architecture of large-scale information system is divided into four layers: security service core, basic security service layer, extended security service layer and security application layer. The previous three layers form the security infrastructure platform shown in Figure 1.

A. Security Service Core

The security service core provides basic and necessary characteristics of general SOA such as service discovering, service searching, security service factory, connection of security service, management of service components, and combination of service and so on. In order to ensure the security of entire security infrastructure platform, the security service core needs to ensure its own security.

In large-scale information system, the design of security service core can adopt traditional distributed technology such as DCOM, RPC, RMI, CORBA, ICE and so on [6, 7]. These

978-1-4244-8625-0/11/$26.00 ©2011 IEEE

techniques are flexible and efficient for remote calls and messages. The design can also apply service-oriented distributed technologies such as Web Service [8], Jini and so on. There are three points need to be considered when using SOA to achieve security service core such as the differences of SOA and distributed objects, the loosely coupled SOA and Coarse-grained services.

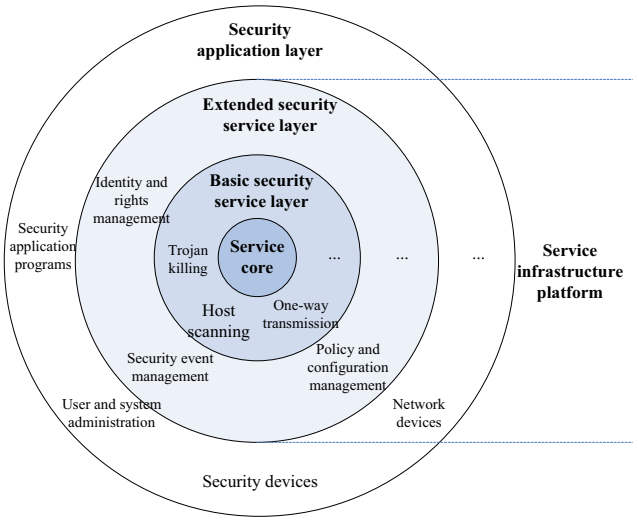

Figure 1. service-oriented security architecture

B. Basic Security Service Layer

The basic security service layer consists of many basic security components which can be dynamically managed in the support of security service core, such as increasing components, removing components and updating components. Each security component should be mapped to one basic security service that cannot be decomposed, such as Trojan killing, Host Scanning, One-way transmission of data.

Each security component can be seen as an agent of specific security service that is provided by security product or technology. When security application makes a call on basic security services, security service core will send service requests to corresponding component. After explaining service requests, security component will control or start corresponding security products and technologies, and will possibly give guidance on manual inspection and policy enforcement. From the SOA design consideration, middleware technology is usually used to realize each security component.

C. Extended Security Service Layer

The extended security service layer is made up a variety of extended security components which can also be dynamically managed in the support of security service core. Each extender security component utilizes security service to find, to search and to apply basic security component. At the same time, through applying composite services and process services to provide more complex security services for outer layer application based on basic security component, such as identity and rights management, security event management, policy and configuration management and so on.

IV. Architecture Design of Security Service Core

A. Key Functions of Security Service Core

The security service core can be seen as communication bus connecting all the service providers and service users. The most important duty is to provide functions of connectivity, data format conversion and service routing for different hardware platforms, software platforms, middleware and protocols. Therefore, these three factors should be considered in designing the architecture of security service core.

B. Choice on architecture design of security service core

There are three patterns can satisfy the functions discussed above: the service connection pattern, the service-driven pattern and the message exchange pattern.

1) service connection pattern

The security service core can use three service connection patterns: direct connection pattern, intermediary pattern and agent pattern. The direct connection pattern means that service user accesses and gets service directly from the specific location of service provider. The intermediary pattern means that security service core acts as one intermediary to encapsulate service calls between service user and provider, and make them couple loosely such as service user applies identity and priority to designate services. This pattern can realize more flexible routing. The agent pattern means to control the access from service user or to service provider by the agents, usually the agent of service provider is enough. The direct connection pattern is suited to small-large distributed system. As the information system gradually develops to cross-domain and cross-application of large distributed system, the intermediary pattern and agent pattern are combined to solve issues of extensibility and scalability. But at the same time, the complexity on design of security service core is also brought in.

2) service driven pattern

The security service core consists of two service driven patterns: protocol driven pattern and low-level interface driven pattern. The protocol driven pattern means that service provider and service user can call and respond services in the light of the unified protocols defined by security service core. The low-level interface driven pattern means that service provider and service user can call directly relevant low interface according to the interface provided by security service core in different platforms (operating system platform and programming language platform).

There are several factors need to be considered in choosing driven pattern of security service core about large-scale information system. Firstly, security service requests are more specific than complex business logic. The protocols of security service core usually need not be changed frequently. Therefore the protocol driven pattern can be used. Secondly, a large number of security devices and technologies are often deployed on various platforms, so the corresponding security service provider will be involved in various platforms. The mapping which is achieved by service provider between specific platforms and protocols can greatly simplify the design of security service core. At last, when adopting security devices

and components designed by third-party, the use of protocol driven pattern can bring more flexibility to developers.

3) message exchange pattern

Services are realized by message exchanging between service provider and service user. There are four message exchange patterns in designing the architecture on security service core: one-way message pattern, request/response pattern, request/callback pattern and publish/subscribe pattern. The one-way message pattern means that message is sent in one direction and sender need not do any work after the message is sent. The request/response pattern is that service user sends a request message and then waits provider answering to the message. The service user is in blocking state during this period. The request/callback pattern can be seen as non-blocking request/response pattern. It means when service user sends request message and simultaneously points out relevant operation after receiving the message. The publish/subscribe pattern means when the message needs to be sent to many receivers, each receiver subscribes the message as an observer. The message will be notified to all observers when it is published.

There are several factors such as the characteristics of security service, efficiency and expansibility need to be considered in choosing message exchange patterns of security service core. If the message needs to be answered, the request/response pattern can be adopted when efficiency and expansibility permit. This can avoid the complexity brought about by the request/callback pattern. For the one-way message, the one-way message pattern and publish/subscribe pattern can be used according to the number of receivers. With the expansion of distributed system, using common transmission path to solve message transmission will inevitably lead to the scalable bottlenecks in distributed system. Therefore, distributed caching need to be designed to improve scalability, efficiency of processing and transmission according to the characteristics of security distributed architecture about large-scale information system.

C. Distributed Cache Design of Security Service Core

The original design aim at SOA is to improve scalability of system through dividing independent application or program library into several parts which can run independently in other server machine. Calls in different services do not maintain any state data [9]. SOA can be easily extended to multiple server machines and across data centers. But in specific applications, complex dependencies among multiple services and frequent calls in network services make the system unsalable. Better design to various applications and services can reduce the effects. In some specific environments, improving transport protocols based on caching mechanism can solve this problem.

In large distributed information system, security components in basic security service layer which are frequently called lead the key point to scalability bottleneck. Each basic security component can be seen as one security agent provided by security device or technology. The services mainly include: to control security device, to start/stop one security technology, to collect the state of security device, to collect security event information, and to call other security business service interfaces. Rational security component design and deployment can reduce the effects that security business service interfaces to system scalability. But security data state and security event information produce large amount of data which update frequently, at the same time need to be received and processed by basic security service layer and extended service layer. Therefore, transmission and processing of security event and security state should be solved in order to realize real scalability. One-way message of security event and security state can adopt distributed cache architecture which can improve the scalability. As shown in Figure 2.

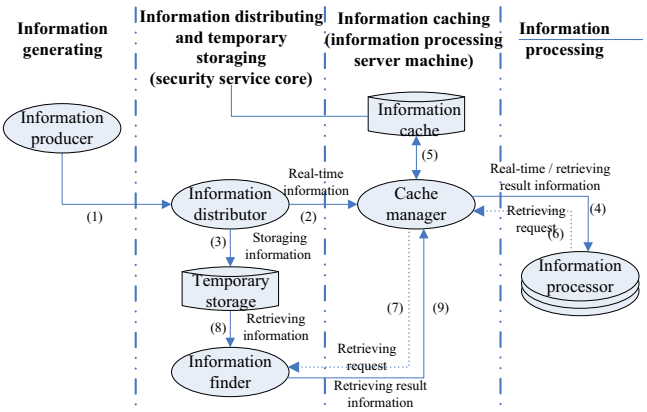

Figure 2. distributed cache architecture

1) The information producer generates real-time informations of security state or security event. The informations enter information distributor of the security service core by using direct connection pattern or intermediary pattern.

2) According to the publish/subscribe pattern, the information distributor dispenses the informations to relevant cache manager via network. The cache manager is located in server of information processor. When starting up information processor, specific type of informations need to be subscribed by cache manager. The cache manager collects all the necessary informations and make a subscribution through information distributor.

3) The information distributor archives the informations to the temporary storage while distributing informations. The temporary storage can store all the informations that have been destined for duration. The file system of rapid retrieving and real-time database should be adopted if the efficiency is considered.

4) According to the publish/subscribe pattern, the cache manager dispenses real-time informations to relevant information processor. As the cache manager and relevant information processor are in the same server machine, inter-process communication can be used to improve transmission efficiency.

5) The cache manger saves informations into the cache while dispensing informations. Therefore, different informaiton processors can retrieve latest informations directly

from the cache and need not access network system. Network load is lower as soon as possible.

6) The information processor sends data retrieval requests through local cache manager when latest informations are needed. If the informations have been received and transmitted by cache manager, all the information processors can get them from cache quickly.

7) The cache manager will send retrieval requests through the information finder which is located in the security service core if the informations fail to be found. The retrieval result is responded when finding the informations. Taking into account the efficiency, the cache manager can return the abstract informations and the informatiom processor can get the informations in cache through inter-process communication.

8) The information finder retrieves the informations that needed to be got back. As the temporary storage adopts the design of rapid retrieval, it can return the retrival information as soon as possible. The design makes different cache managers in the whole system get latest informations from temporary storage directly and need not access specific storage system.

9) After receiving the retrieval informations that are sent back by information finder, the cache manager needs to transmit the informations for information processor and store the informations into the cache.

V. CONCLUSIONS

This paper makes a deep study on security problems of large-scale information system. The service-oriented security architecture and the architecture of security service core are proposed in turns. The design has certain guidance to the security system construction of large-scale information system. But there are still some details that need to be considered in practice.

REFERENCES

[1] P. Feiler, R. Gabriel, J. Goodenough. “Ultra-Large-Scale Systems. The Software Challenge of the Future”. Software Engineering Institute. Carnegie Mellon University. June 2006.

[2] C. Ghezzi, “Service-Oriented Computing: Where Does It Come From? A Software Engineering Perspective,” keynote address, 3rd Int’l Conf. Service-Oriented Computing, Amsterdam, Dec. 2005.

[3] Thomas. Erl. “SOA: Principles of Sevice Design”, Pearson Education inc, 2008.

[4] Nicolai M.Josuttis. “SOA in Practice: The Art of Distributed System Design”, O’Reilly Media inc, 2007.

[5] M. P. Papazoglou, W. J. Heuvel.“Service-Oriented Architectures: Approaches, Technologies and Research Issues,” VLDB J, vol. 16, no. 3, 2007, pp. 389-415.

[6] Sumair Khan, Kalim Qureshi, Haroon Rashid. “Performance Comparison of ICE, HORB, CORBA and Dot NET Remoting Middleware Technologies”. International Journal of Computer Application, 2010,vol. 3, pp15-18.

[7] SeongKi Kim,Sang-Yong Han. “Performance comparison of DCOM, CORBA and Web Service”. In PDPTA, pp 106-112, 2006.

[8] M.P. Papazoglou. “Web Services: Principles and Technology”, Prentice Hall, 2007.

[9] M. Arrott, B. Demchak, V. Errnagan. “Rich Services: The Integration Piece of the SOA Puzzle”. In: Proceedings of the IEEE International Conference on Web Services (ICWS 2007), Salt Lake City, USA, pp. 176-183. IEEE Computer Society Press, Los Alamitos, 2007.