ISSN 1009 - 3095 *Journal of Zhejiang University SCIENCE* V.3, No.1, P.30 - 36, Jan. - Mar., 2002
http://www.periodicals.com.cn; http://www.zju.edu.cn/depart/zdxb

http://lib.zju.edu.cn/eindex.htm; http://www.zjupress.com; jzu_s@mail.hz.zj.cn

A reference model for database security proxy*

CAI Liang(蔡 亮), YANG Xiao-hu(杨小虎), DONG Jin-xiang(董金祥)
(*Artificial Intelligence Institute, Zhejiang University, Hangzhou 310027, China*)

Received Dec.15, 2000; revision accepted June 6, 2001

Abstract: How to protect the database, the kernel resources of information warfare, is becoming more and more important since the rapid development of computer and communication technology. As an application-level firewall, database security proxy can successfully repulse attacks originated from outside the network, reduce to zerolevel damage from foreign DBMS products. We enhanced the capability of the COAST's firewall reference model by adding a transmission unit modification function and an attribute value mapping function, describes the schematic and semantic layer reference model, and finally forms a reference model for DBMS security proxy which greatly helps in the design and implementation of database security proxies. This modeling process can clearly separate the system functionality into three layers, define the possible security functions for each layer, and estimate the computational cost for each layer.

Key words: information warfare, database security, firewall, reference model, security proxy
Document code: A **CLC number:** TP309.2

INTRODUCTION

In general, the analysis, manipulation, and simulation of a modeled system can lead with risk to new knowledge and insight on cost, or inconvenience associated with its direct manipulation. The process of modeling a system gives the modeler an improved understanding of the modeled system. One of the limitations system developers face is their own inability to cope simultaneously with too many details reference models help to overcome this limitation.

This paper introduces a reference model for database security proxy. It enhances the capability of the firewall reference model suggested by Purdue University's COAST Laboratory by adding a transmission unit modification (TUM) function and an attribute value mapping (AVM) function; describes the schematic and semantic layer reference model, and finally forms a reference model for DBMS security proxy which is of great help in the design and implementation of database security proxies.

The Artificial Intelligence Institute of Zhejiang University has prototyped a Database Security and Protection System (DSPS) to counter possible malicious behavior of untrusted DBMS. Based on the malicious DBMS threat model, DSPS can greatly improve the critical application's capability to cope with malicious DBMS by incorporating self-controlled authentication, principal/object mapping, transparent attribute encryption/decryption, attribute/tuple level mandatory access control, and parallel structure of multiple DBMSs.

DSPS runs as a security proxy to the untrusted back-end DBMS. According to its logical position in the network, it is essentially an application-level firewall. Besides the common capability of repulsing some attacks from outside the network, it can enforce some more database-specific security polices in much more granularity using the knowledge of the schematic and semantic information of the back-end database. There are three kinds of functions can be integrated into the database security proxies:

1. Preventing some attacks from outside the network
2. Enforcing some new security polices, and in much more granularity
3. Enhancing the system capability for repulsing the malicious DBMS

* Project(No.45.6.1-017) supported by the State Defense Scientific Industry Committee of China.

However, the firewall product development is only an engineering process. There isn't any theoretic background for firewall technology, let alone the application-specific database security proxy(Ioannidis et al., 2000). In order to promote the assurance level of the security products and give the developers an improved understanding of the system, it is necessary to establish a reference model for database security proxy. Through this modeling process we can clearly separate the system functionality into three layers, define the possible security functions for each layer, and estimate the computational cost for each layer.

INTRODUCTION TO COAST FIREWALL REFERENCE MODEL

The firewall reference model suggested by Purdue University's COAST Laboratory (called FWRM for short) is a general layered model for firewall technology. It summarizes the basic security functions for network access control (authentication function, integrity function, access control function, audit function, and access enforcement function) and is of great help to the design of network firewalls (Christoph et al., 1997).

Fig. 1 displays the single layer FWRM. Consider the case where a principal A outside of a protected network security policy domain attempts to communicate with a principal B inside that domain. The gap between "Out" and "In" can be filled with intermediate networks of any technology and topology so long as data can be transmitted between the sender's and the receiver's networks. Everything between the gap and the representation of principal B is considered part of the protected network policy domain.

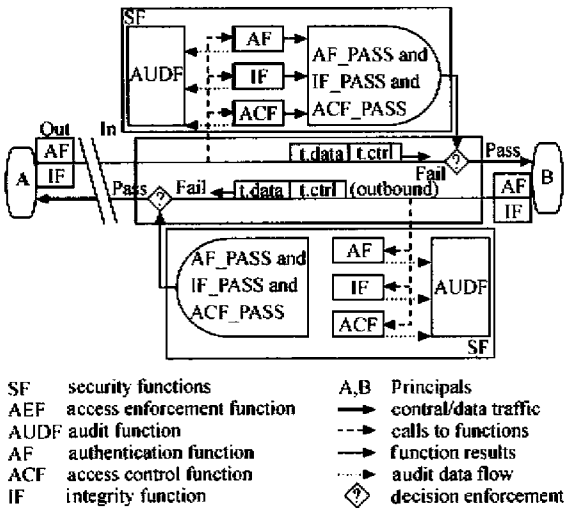

Fig. 1 The single layer COAST firewall reference model

Shaded boxes represent functions. The boxes labeled SF represent a collection of security functions that are applied to transmission units exchanged between principals A and B. The dashed arrows represent the invocation of this collective function SF. Each SF receives portions or possible even (a copy of) the entire transmission unit as input arguments. SFs calculate a result PASS or FAIL for each transmission unit. The diamond with the question mark represents the matching of the decision to its transmission unit and the decision branching and enforcement depending on the result. If the result is PASS, the transmission unit is forwarded to its destination; if the result is FAIL, an exception occurs, and the transmission unit is dealt with accordingly (e.g., recorded to the audit log, and then discarded). The separation of SF into two boxes serves to further illustrate the bi-directionality of communications.

The access enforcement function (AEF) located in the communication path between these two principals may request the authentication of each transmission unit, the verification of the integrity of each transmission unit, the access control decision, and enforce the results of these functions.

The authentication function (AF) is used to assure that the transmission units' apparent and actual origins are identical. The integrity of the transmission units can be verified by the integrity function (IF). The access control function (ACF) determines if the transmission unit is to be forwarded further towards its destination. This decision is based on the control information (t.ctrl) and data information (t.data) in each transmission unit.

Arrows with thin, dashed lines indicate possible invocations of the audit function (AudF). All blocks that are part of the firewall system have invocation access to the audit function to record events and data according to the network

domain security policy in force.

For any network transmission unit, functions AF, IF, and ACF can be called in any order. Their results are combined into a final result using logical operation "AND". If a single function generates FAIL as a result, the transmission unit should not be forwarded to its destination.

As shown in Fig. 1, outbound communication traffic is subject to the same security functions as inbound communication traffic. However, because of the trust relationship between the firewall and internal principals, it may not be necessary to enforce the same functions as inbound traffic. For example, if a "trust relationship" exists between internal hosts and the firewall, a firewall designer may choose to omit outbound authentication verification of communication traffic.

The dashed boxes with labels AF and IF close to principals A and B indicate cooperation by a sender for the authentication and integrity functions. Cryptographic protocols, the primary means in network security to provide authentication and integrity assurance services, may require the participation of the sender (e.g., to provide cryptographic secrets for the generation of session keys). Without this cooperations, cryptographic protocols could not be used to provide the necessary services of AF and IF. The box is dashed to indicate that they are authentication procedures that do not require participation of the sender, and to represent that the participation is not under the control of the firewall.

LAYERED REFERENCE MODEL FOR DATABASE SECURITY PROXY

The reference model for database security proxy enhances the capability of the firewall reference model suggested by Purdue University's COAST Laboratory by adding the transmission unit modification (TUM) function and the attribute value mapping (AVM) function. According to the characteristics of database applications this model clearly separates the system functionality into three layers, defines the possible security functions for each layer, and estimates the computational cost for each layer. The layered model is of great help in the design and implementation of database security proxies.

In this section we first introduce the criteria used to separate the database security proxy into three layers, then put the emphasis on the system components and the interaction between components of the schematic and semantic layer. Limited by the paper size only newly added components are discussed here.

Separation criteria

A database security proxy can be logically separated into three layers according to the different knowledge, computation target, and the control granularity of each layer. The properties of these layers are summarized in Table 1.

Table 1 The separation criteria of the reference model for database security proxy

Layer	Network layer	Schematic layer	Semantic layer
Known knowledge	Network protocols	Schema of the back-end database	Semantic of the back-end database
Computational target	Protocol header of the transmission unit	Schematic information in the request and response	Each attribute value in the request and response
Control granularity	Protocol attributes	Schema objects	Object values
Control cost	Low	Medium	High
Reference model used to describe this layer	COAST's FWRM	Schematic layer of application level firewall reference model	Semantic layer of application level firewall reference model

1. Network layer provides the ordinary functionality of the network firewall. It is oriented to various network transmission protocols and enforces the security policies according to the information contained in the header of transmission unit. Therefore, the knowledge known to this layer is various network protocol definitions, the computation target of this layer is the protocol header of the transmission unit, and the control granularity which can be attained by this layer is

determined by the granularity of the protocol attributes. This layer can be easily illustrated by the FWRM.

2. Schematic layer enforces the security polices related to the database schema. Besides the database schema, it understands the communication protocol used by DBMS. Therefore, it can extract and analyze the schema-related information from the communication traffic between DBMS and its clients. The computation target of this layer is the schema information contained in the SQL commands and the responses. The control granularity can be as fine as the granularity of the database schema including various kinds of database objects, object names, and object properties (such as the table name and the attribute name).

3. Semantic layer enforces the security polices related to the semantics of each data item. Besides the database schema, it knows part of the semantics information. Thus it can understand and analyze the semantics contained in almost all the communication traffic between DBMS and its clients. It can enforce the access control according to the specific values of every attribute and transparently modify the communication traffic by the transmission unit modification (TUM) function. That is the finest granularity that could be attained by database security proxies. When compared with the schematic layer the semantic layer is "value-oriented", whereas the schematic layer is "name-oriented".

Schematic layer reference model

The schematic layer reference model is illustrated in Fig. 2. Note that the database security proxy has already interpreted the application-level protocol in the schematic layer and above. So the principal B in the network security policy domain of the FWRM is replaced to be a more specific one: DBMS. Further more, the asymmetry of the interaction between DBMS and its clients results in the different approaches used to process the inbound and outbound communication traffic.

To all the inbound traffic, schema parsing function is responsible for extracting the schema-related information from the application-level transmission unit. Within all these schema-related information the principal, object, and action are particularly important.

In order to reduce the possible information leakage by principal a some mission-critical applications require that the DBMS clients (the principal A in Fig. 2) run in the untrusted environment know as little as possible schema information. Therefore, after the schema parsing function there are three functions named PM, OM, and AM responsible for principal mapping, object mapping and action mapping respectively. The PM, OM and AM are equal to the invariant mapping if there is no need to confine the schema information. Otherwise, after these mappings the real database schema which is of rich semantics will be confined between the database security proxy and the back-end DBMS, whereas the schema known to the clients (which are outside the database security proxy) has already been obfuscated and is very hard to understand. For example, the PM maps the principal Account1 into Alice; OM maps the object Table1 into Missile-tab, Attribute1 into Missile-Model, and Attribute2 into Missile-Accuracy; and AM maps the action Action1 into SELECT. Now, if the principal Alice outside the network security protection domain wants to execute the SQL command "SELECT Missile-Model, Missile-Accuracy FROM Missile-tab", she must get the identity Account1 from the database security proxy (that is a chance for us to enforce self-controlled authentication), and then issue the command "Action1 Attribute1, Attribute2, Table1" to the database security proxy. After applying the PM, OM and AM the command is translated into "SELECT Missile-Model, Missile-Accuracy FROM Missile-tab" and sent to the back-end DBMS.

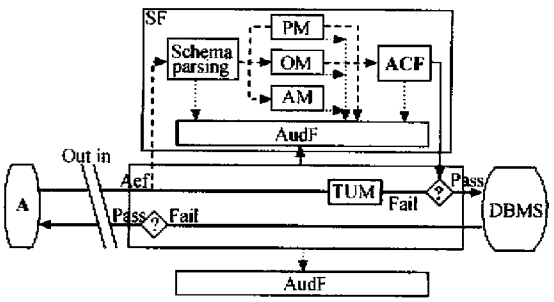

Fig. 2 The schematic layer reference model for database security proxy

It is very interesting that if we replace the PM and OM mapping functions with their inverse

functions the schema information will be confined between the clients and database security proxy. In the above example, the principal still issues the normal SQL command "SELECT Missile-Model, Missile-Accuracy FROM Missile-tab". After applying the PM and OM the command is translated into "SELECT Attribute1, Attribute2 FROM Table1" and sent to the back-end DBMS. In this way the semantic information contained in the real database schema is reduced to be minimal, that is the key idea used to prevent the untrusted DBMS from triggering the malicious code according to the semantic information in the database schema (For example, the DBMS automatically steals or corrupts the data items on seeing the table whose name includes Missile).

The transmission unit modification function (TUM) is the most important enhancement we add to the COAST's firewall reference model. It sits on the communication path, and is capable of modifying the inbound and outbound transmission unit. After incorporating the TUM the firewall reference model not only could determine the PASS or FAIL of each transmission unit, but also could modify the transmission unit according to the security policy. That will greatly improve the capability of the firewall and make it diversified. To the database security proxy, the application-level firewall, the TUM is more important. In combination of the TUM and the known application semantics the database security proxy can break through the category of enforcing new access control policies on the legacy application and be promoted into the category of adding new security properties into the legacy application.

In the schematic layer reference model the TUM is responsible for enforcing the results of PM, OM and AM on the inbound traffic so that the traffic which reaches the DBMS has already been changed to comply with the security policies.

As in the COAST's firewall reference model ACF will determine if the transmission unit is to be forwarded further towards its destination. This decision is based on the mapped principal, object, action, access context and environment information. There are two kinds of access controls here. One is to enhance the access control provided by the back-end DBMS, such as the access control in much finer granularity, context-based access control, history-based access control, attribute-level mandatory access control and so on. The other is to enforce self-controlled access controls required by some mission-critical applications which do not trust the foreign DBMS products (the access controls enforced by database security and the DBMS are working concurrently).

Because the response from DBMS normally is a set of attribute values which belong to the semantic category, there is no specific security control function applied to the outbound traffic in the schematic layer reference model.

It must be emphasized that the schematic lay reference model (including the semantic layer reference model described in the next subsection) is kept at a high level of abstraction to concentrate on the information flows and functional dependency of its components. The representation of the model is independent of its implementation. For example, in order to improve the performance the TUM is always called after the ACF in typical implementations. In this scenario only the transmission units which pass the access control need to be modified.

Semantic layer reference model

The semantic layer reference model is illustrated in Fig. 3. In contrast to the schematic layer, the semantic layer puts its emphasis on the outbound traffic control.

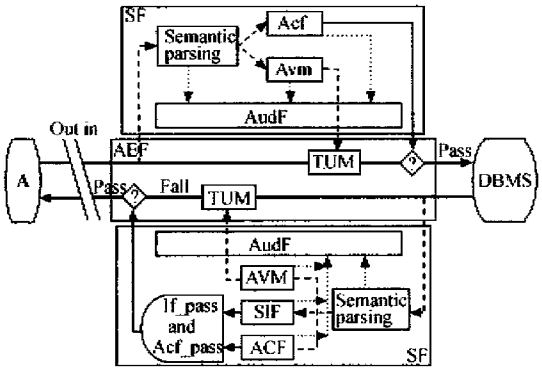

Fig.3 The semantic layer reference model for database security proxy

The semantic information contained in the inbound traffic (attribute value) is mainly used as the constraints of the WHERE clause. Semantic parsing function is responsible for extracting them from the application-level protocol data

units. As shown in Fig.3, the semantic information can be used in two ways:

1. One is to be used as the parameters of ACF, so as to support the semantic-related access control policies, such as prohibit the principal Alice from accessing the tuples in Missile-tab where the attribute Missile-Model equals '10A'. Although the back-end DBMS can enforce the same access control by views, there are some special application environments where the security controls provided by the back-end DBMS are not trusted. Moreover, some semantic-related access controls which are enforced by database security proxy cannot be provided by the ordinary commercial-off-the-shelf DBMS products. For example, the security policy requires the enforcement of semantic-based MAC labeling (The tuples in the table are classified to different sensitivity levels according to the values of one or more attributes.) for the Missile-tab. The tuples whose attribute Missile-Model equals '10A' are classified to be confidential, the one that equals '10B' is secret, and the one that equals '10C' is top secret.

2. The other is to be used as the parameters of attribute value mapping (AVM) function, so as to provide the transparent attribute encryption function or deliberately obfuscate the semantics of some attributes values. AVM is responsible for mapping the attribute value from one to another, and is the most important security functions introduced in the semantic layer reference model. A typical application of AVM is to provide the attribute encryption/decryption function which is transparent to the database applications. In this scenario the AVM function on the inbound traffic is an encryption function, and the one on the outbound traffic is the corresponding decryption function. Therefore, some sensitive attribute values are stored as the ciphertext in the back-end database. This can prevent some attacks that originate from the system console, such as directly getting the sensitive values using the SQL command, getting them through database dump, directly copying the database files through the operation system, and so on. Besides the encryption and decryption function, AVM can also utilize some special mathematical transformations to deliberately obfuscate the semantics of the attribute value (something similar to the PM, OM and AM functions in the schematic layer reference model). The results of AVM are enforced onto the inbound and outbound traffic by TUM.

To the database security proxy, almost all the information contained in the application-level protocol data units of outbound traffic (such as the execution status of each command and the attribute values return by the SELECT command, and so on.) is semantic information. So there are richer security functions that can be enforced on the outbound traffic than the ones on the inbound traffic. Besides the semantic parsing, AVM and ACF which are the same to the inbound security functions, there is a very important security function, named super integrity function (SIF), which is applicable to the outbound traffic. We deliberately separate the SIF from the IF used in the COAST's firewall reference model, because the SIF can integrate some "super" integrity control functions, including the integrity-lock and the parallel structure of multiple DBMSs, based on the semantic information known to this layer.

The integrity-lock architecture is an approach to construct a multilevel secure database management system using existing DBMS. Tuple-level mandatory access control can be enforced using this architecture. A cryptographic checksum is attached to each tuple. This cryptographic checksum is stored in database as a transparent attribute, and it is formed by the content of this tuple, its physical location and its security label. Before returning data to the principal a, SIF will verify the cryptographic checksum to ensure that each tuple and its label have not been tampered with and ACF will discard tuples that do not pass the mandatory access control policy. For detailed information about integrity-lock, please refer to references(Graubert et al., 1985; Denning, 1985).

In the information warfare we should repulse possible malicious behaviors of the untrusted DBMS products. We found that the malicious behavior which is the most difficult to detect is to deliberately misinterpret the SQL commands issued by the principal (although we can assure the validity of UPDATE, INSERT instructions by enforcing the integrity lock architecture.):

1. There are N tuples conforming to the WHERE clause of SELECT command in the database, but the malicious DBMS deliberately re-

turn more or less tuples.

2. There are N tuples that should be deleted according to the WHERE clause of DELETE command, but the malicious DBMS deliberately delete more or less tuples.

This deliberately misinterpretation of the operation semantics is the most effective attack in information warfare. Usually the M/N is very small, so these malicious behaviors are hard to detect. With time going on this erroneous semantics will contaminate the whole database like an infectious disease. In information warfare the erroneous semantics will result in erroneous decision and losing the battle. Moreover, even if these kind of malicious behaviors are detected, the traditional database recovery mechanism is of limited use for recovering the database because the users, applications and the DBMS themselves participate in the propagation, feedback, and dissemination process of the erroneous semantics(Paul et al., 1997).

SIF can incorporate the parallel structure of multiple DBMSs to repulse these attacks. Based on the logical replication technology (McDermott, 1997), the parallel structure can be used to detect the possible Trojan Horses in the untrusted back-end DBMS. In this structure, each SQL command will be executed in multiple DBMSs, and the execution results will be compared. If they differ, there is at least one DBMS which had maliciously behaved.

CONCLUSIONS

With more and more firewall products being developed and their effects being proved, a lot of researchers dedicated themselves to establishing a sound theoretic foundation for the design and implementation of firewall products. The general reference model proposed by Purdue University's COAST Laboratory is one of the representatives. It summarizes the basic security functions for network access control and is of great help in the design of network firewalls.

This paper introduces a layered reference model for database security proxy. It enhances the capability of COAST's firewall reference model by adding the transmission unit modification (TUM) function and an attribute value mapping (AVM) function, and describes the schematic and semantic layer reference model. A database security proxy can be logically separated into three layers (network layer, schematic layer and semantic layer) according to the different knowledge, computation target, and the control granularity of each layer.

Compared with the results of other related researches, our reference model has the following characteristics:

1. It is an application-level firewall model, especially for database applications.

2. It provides the security functions for repulsing malicious DBMS in information warfare.

3. It is a layered model which clearly defines the possible security functions for each layer, and estimates the computational cost for each layer.

Based on the malicious DBMS threat model, we prototyped a Database Security and Protection System (DSPS). It can greatly improve the critical application's capability to repulse malicious DBMS by incorporating self-controlled authentication, principal/object mapping, transparent attribute encryption/decryption, attribute/tuple level mandatory access control, and parallel structure of multiple DBMSs. currently, the DSPS is under testing in a Chinese defense department.

References

Christoph, L.S., Eugene, H.S., 1997. A reference model for firewall technology. Proceedings of the 13th Annual Computer Security Applications Conference (ACSAC), p.133 – 145.

Denning, D., 1985. Commutative filters for reducing inference threats in multilevel database systems. Proceedings of IEEE Symposium on Security and Privacy, p.134 – 146.

Graubert, R., Duffy, K., 1985. Design overview for retrofitting integrity-lock architecture onto a commercial DBMS. Proceedings of IEEE Symposium on Security and Privacy, p.147 – 159.

Ioannidis, S., Keromytis, A., Bellovin, S., et al., 2000. Implementing a distributed firewall. Proceedings of Computer and Communications Security, p.190 – 199.

McDermott, J., 1997. Replication does survive information warfare attacks. Proceedings of IFIP WG 11.3 Annual Working Conference on Database Security, p. 219 – 228.

Paul, A., Sushil, J., Catherine, D., et al., 1997. Surviving information warfare attacks on databases. Proceedings of IEEE Symposium on Security and Privacy, p.164 – 174.