SECURE DATABASE CONNECTIVITY ON THE WWW

Marijke Coetzee
School of Information Technology
Technikon Witwatersrand
South Africa
coetmj@mweb.co.za

Jan Eloff
Department of Computer Science
Rand Afrikaans University, Johannesburg,
Auckland Park, South Africa
eloff@rkw.rau.ac.za

Key words: Information security, Internet, database server, web server, application server

Abstract: The rapid growth of the Internet increases the importance of connecting to existing databases. The Web, with all its versatility, is putting database security to the test. Access to web-enabled databases containing sensitive information such as credit card numbers must be made available only to those who need it. The focus of this paper is to shed some light on how databases can be used in a secure manner when connecting to the World Wide Web, by investigating the application of current state-of-the-art database security services.

1. INTRODUCTION

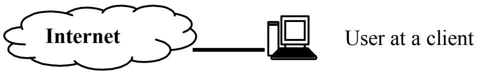
Information can be considered a company's most valued asset and should be protected as such. In the past, companies allowed very limited access to corporate information. Today, the incidence of companies making their corporate information available to remote users, through the Internet, is ever-increasing. Access control over this information should be carefully exercised in order not to hinder the user navigation experience.

Users are given access to corporate information through web-enabled databases. A web-enabled database requires three essential components: a web server, an application server and a database server. This can be called a virtual web database environment. This environment is shown in figure 1.

It is important to first distinguish between the functionality of each of the three components of the virtual web database environment.

The first component of the virtual web database environment is the *web server*. The main function of the web server is to process presentation logic. Web pages are delivered to clients in response to requests for URLs.

An *application server* in a virtual web database environment receives requests from the web server, runs the business logic and provides connectivity to other application servers or database servers.



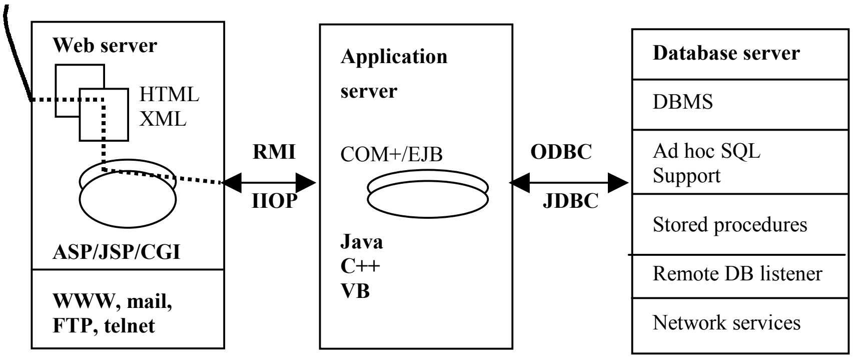


Figure 1: A virtual web database environment

A *database server* in a virtual web database environment is a repository for all information. It receives requests from the application server for access to information in the database server through data access technologies such as ODBC or JDBC.

The security of the virtual web database environment is challenged, as huge user populations access corporate information, past traditional perimeters. Examples of virtual web database environments are online banking where customers access their accounts via the Internet, and e-commerce merchants that trade with customers anywhere in the world. Since most malicious intrusions occur from inside, defences such as firewalls, intrusion detection and virus scanning provide limited protection.

In this new and evolving environment, database security has been a well-researched topic ever since the first databases were used. Over the years, a substantial body of knowledge of database security issues has developed. Traditional database security could therefore provide a basic framework to be used when approaching the security of the virtual web database environment.

The remainder of this paper will concentrate on some current database security services and mechanisms to determine how and to what extent they can secure a virtual web database environment.

2. DATABASE SECURITY SERVICES

Database security comprises a set of services and associated mechanisms that can provide protection to information. Physical issues such as the protection of equipment and facilities, logical issues such as access control models and organizational issues such as manageability are usually addressed.

Most commercial DBMSs (database management systems) are built with the Trusted Subject architecture, [CAST95] where a trusted DBMS plays an important role in enforcing some security services. Table 1 lists identified database security services. A definition of the service is provided, as well as the associated mechanisms that can be used to enforce the service.

Table 1: Database security services and associated mechanisms

1. Identification and authentication

The ability to uniquely identify all database subjects such as users or programs Authentication performed by the database, operating system and others such as DCE (Distributed Computing Environment), Kerberos, or trusted clients
2. Authorization *The ability to control the actions of identified database subjects* Discretionary access control (DAC), mandatory access control (MAC) and role-based access control (RBAC) policies
3. Confidentiality *The ability to prevent the improper disclosure of data* Encryption of stored information, inference control and employee confidentiality training
4. Integrity *The ability to ensure the validity of data – this consists of 3 aspects:* ***4.1*** *Database integrity – the ability to ensure the integrity of stored data and software.* Checksums, virus and Trojan Horse protection, removal of software vulnerabilities, configuration of database software *4.2 Operational integrity – the ability to ensure the integrity of database transactions* Transaction processor *4.3 Semantic integrity – the ability to ensure the integrity of the value of data* Entity integrity, referential integrity, domain integrity, user-defined integrity, normalization
5. Accountability *The ability to ensure that database subjects can be held accountable for their actions* Audit logs
6. Availability *The ability to ensure that information is accessible when needed* Hardware redundancy, for instance RAID, backup/recovery, replication/partitioning of tables, contingency plan
7. Manageability *The ability to easily manage the security of the database* Security management tools
8. Assurance *The ability to determine the degree of confidence to which the security needs of the database are satisfied* Certification with ITSEC or TCSEC, test security configuration
9. Physical security *The ability to prevent unauthorized access, damage and inferences to the database* Secure buildings and equipment

3. DATABASE SECURITY SERVICES APPLIED TO THE VIRTUAL WEB DATABASE ENVIRONMENT

To determine the measure of security that can be provided by current state-of-the-art database security services, their implementation must be investigated at each of the three servers of the virtual web database environment. Since the same security service is applied at each server, the influence of the dissemination of a service needs to be determined. This can then be contrasted to the service as provided by database security in the conventional context. As the authentication and authorization services require careful integration, they will be discussed in more detail.

3.1 IDENTIFICATION AND AUTHENTICATION

The basis of a security system is the correct identification of subjects. Table 1 shows that the DBMS or operating system can identify and authenticate subjects, mostly with a password. Subjects are given a security context, against which all subsequent requests in a session will be evaluated. A trusted path is required to ensure that subjects are not “spoofed” when communicating with the security system. [PFLE97]

Identification and authentication of the virtual web database environment

A virtual web database environment increases the complexity of user identification and authentication. If identification and authentication are enforced at each server of the virtual web database environment, it might turn users away.

The *web server*, an untrusted application, is the first point of contact that a user makes with the virtual web database environment. Although it is known to be insecure [FRAN01], the web server can be configured to perform this service. For requests that need authentication, traditional username and password pairs can be used, encrypted with SSL (Secure Sockets Layer) [SSLI01]. Other more sophisticated methods such as digital certificates, secure cookies [PARK00] or electronic cards can be employed to enhance authentication. To ensure that a user is not "spoofed" into communicating with an impostor, the client can authenticate the web server with a digital certificate. This allows a form of "trusted path" to be created between the user and web server.

Furthermore, a secure session must be maintained between the user and the web server, and beyond. As HTTP is a stateless protocol, it does not remember user interactions from request to request. The web server must maintain secure session state for a user, when a number of pages are requested in succession, to prevent the user from re-authenticating for each request. This is achieved by storing session details such as unique session IDs in digitally signed cookies at the client, or in an external database. Web server processes, replicated to enhance performance, will be a further complication in the creation of a secure session.

Once the web server has authenticated the user, it can pass the validated user ID to the *application server* and the user need not be authenticated again. The application server can only accept these user credentials if a relationship of trust is established between the web server and application server with digital certificates. The secure session state needs to be further maintained for a user interaction, as various application server components, residing on one or more application server instances, are invoked in turn.

To complete the user transaction, data in the *database server* will be read or updated. Ideally, the database server should authenticate each user. This, however, is not practical in the case of a virtual web database environment where millions of users may be accessing the database server. By allowing the application server to act on behalf of users, this problem can be overcome. Therefore, the application server is delegated an appropriate degree of trust by the database server as it presents a digital certificate.

The responsibility of web server identification and authentication of users should be clear, since all other servers will trust it to perform this service. If the web server fails to perform this service, any debate over access control policies become irrelevant. [WISE01] The implementation of this service in a virtual web database environment can be vastly improved if operating system or directory services such as LDAP are used. This allows the creation of a security context, against which further requests can be evaluated.

It is clear that this service cannot be implemented to the same level of assurance as in database security. As identification is filtered through to the database server, the identity of the real user is lost and is replaced by that of the application server. The service is performed by various unreliable applications that have to be integrated with each other. Any small error can defeat a secure session and can allow a malicious user to pose as another.

3.2 AUTHORIZATION

Database security provides a complete set of authorization policies in the form of DAC (discretionary access control), MAC (mandatory access control) and RBAC (role-based access control), as shown in table 1. Fine-tuned access control to related database objects is centrally administered by the DBMS.

- With DAC, a subject may own database objects and may have the discretion to grant others access to those objects. Each subject or group of subjects must be assigned a set of permissions.
- With MAC, access control is beyond the control of the subject. All access control is set by the administrator, based on labels such as "secret", assigned to all subjects and objects.
- RBAC allows the creation of roles, to which all permissions are assigned. Since subjects are assigned roles and roles are created according to the organization structure, it is a more manageable policy.

These authorization policies need to be evaluated to determine the measure of security that they can provide to the virtual web database environment, as well as the ease with which they can be implemented. Their application will also depend on whether they are supported by the various servers.

Authorization of the virtual web database environment

Not all authorization policies may be appropriate to use in the virtual web database environment. Even though each server in the virtual web database environment can support DAC, the large numbers of users, HTML pages, scripts, methods exposed by components and database objects make this policy difficult to maintain. Furthermore, DAC allows data to be copied to unauthorized users. For this reason, DAC might be unsuitable to use in a virtual web database environment, where a high level of assurance is required.

As the flow of information can be controlled, MAC provides assured access control. [JOSH01] A complication is that it would be very difficult to label all users, components and database objects in a commercial environment. The administrator must do all labelling, and any small mistake can defeat the policy.

RBAC allows simplified security management since permissions are associated with roles and not individuals. It can therefore directly support the organization-specific policies. As it is policy-neutral, RBAC can be configured to support both DAC and MAC. [OSBO00]

In the virtual web database environment, authorization policies need to be implemented at each server to control what users may see and do.

The *web server* supports DAC. By grouping users into groups such as "customer" and assigning permissions to these groups, the web server controls access to web server resources such as static HTML files and ASP or CGI scripts, with substantial administration overhead. Blocking specific IP addresses, or using non-default port numbers can further limit access. Access is granted or denied to the front-end of an application at web server level by granting access to a specific web page.

As DAC is problematic, efforts have been made to improve web server authorization. RBAC at the web server, RBAC/Web, has been reported [FERR99], but is not common. Several other RBAC implementations have been developed, such as TrustedWeb and getAccess.

From there, RBAC becomes central to access control in the virtual web database environment.

Application servers can employ both DAC and RBAC as access control policies. RBAC would ideally be used to enforce access control at the application server since the permissions granted to roles are usually related to what actions the users are allowed to do. [SAND94] As application servers provide services that allow administrators to completely configure RBAC to applications or methods exposed by components, access control can be highly assured.

It then follows that role-enabled application servers allow application components to deliver customized content to users through the web server interface. This is be achieved by associating web server groups and application server roles with each other, through operating system or vendor specific LDAP directories.

Finally, the role enabled at the application server is filtered through to the database server. RBAC, with roles such as “customer”, can be used to simplify assigning privileges to application servers acting on behalf of users. The privileges that this database role will possess should not be more than what the user, for whom the application server is making the request, should have. The use of views and stored procedures can further enhance access control by not allowing direct access to tables or parts of tables.

It is clear that RBAC plays an important role in ensuring manageable access control to the virtual web database environment, with a very large user population. Both the application server and database server directly support roles. This highlights the need for RBAC at the web server.

Compared to the centrally administered access control of the DBMS, through a single reference monitor, the access control of the virtual web database environment needs careful integration and maintenance by administrators at each server as to appear seamless. This process could be error-prone and lead to improper access control. Integration with enterprise directories containing users, groups, permissions and roles will greatly enhance authorization.

3.3 CONFIDENTIALITY

Database security provides confidentiality as a service to prevent the improper disclosure of information stored in the database. Using encryption of fields, rows, tables or databases, inference control and employee training, as shown in table 1, enforces this service.

Confidentiality of the virtual web database environment

Sensitive information, such as credit card numbers, is transmitted to the web server over open public lines. If electronic snoops were to “eavesdrop” on this connection, they would be able to copy every byte of information. In such situations, the message from the client to the web server, application server and database server needs to be protected against unintended disclosure with SSL.

Confidentiality of some sensitive fields, records or tables stored in the database server can be encrypted to protect it from disclosure. The web interface, which might allow users to run database queries, must not allow sensitive information to be inferred from non-sensitive information. Finally, employees should be sensitized to not disclose sensitive information to unauthorized people.

This database security service can provide a good measure of protection to sensitive information.

3.4 INTEGRITY

The aim of database integrity is to protect the validity of stored data. This is done by ensuring the integrity of database server software, through checksums on stored data, semantic integrity constraints and atomic transactions, as shown in table 1.

Integrity of the virtual web database environment

The database server will carry most of the responsibility of data integrity in the virtual web database environment by implementing mechanisms as stipulated by database security. If the virtual web database environment processes a high volume of transactions that may possibly span across more than one database server, the responsibility of ensuring atomic transactions may have to move to the application server.

The combination of the insecure medium on which the virtual web database environment is run, and the untrusted software that is used adds another dimension to the integrity of the virtual web database environment.

Most security breaches occur as a result of the loss of integrity at the web server. If the web server allows a hacker to gain administrative control through security vulnerability, no measure of data integrity can protect the information in the database server. It is therefore of prime importance to protect the integrity of all web server pages and software by integrity checks run at regular intervals. This can also ensure that no invalid content is delivered to customers. In the same way, the integrity of application server software and components can be assured. All software must be actively configured and maintained, so as not to allow virtual web database environment exploits.

In addition, the integrity of information moved across the communication lines must be protected with SSL through its message authentication code (MAC).

The virtual web database environment is a complex environment and consists of various types of software, providing configurable functionality. Protecting the integrity of the software of each and every server, all HTML pages and scripts, application server components, all database objects, as well as messages transmitted between severs will not be an easy task. Compared to the centrally managed integrity constraints enforced by the DBMS in database security, the integrity of the virtual web database is difficult to achieve and maintain.

3.5 ACCOUNTABILITY

Database security provides accountability as a service that is implemented with detailed audit logs, recording security related events at record, field and element level. Audit records are maintained and protected, allowing threat detection and accountability.

Accountability of the virtual web database environment

Each server in the virtual web database environment has the ability to create its own set of audit logs.

Web server logs show all successful and unsuccessful accesses, with a high level of redundancy. Custom logging is often performed at the application servers, showing how components in a user interaction are invoked. Database servers can perform detailed audit as information is accessed, but as this can be a burden to the system, it is often not used, or it is used in such a way as to not be meaningful.

Ideally, audit should record all actions of a user as a transaction is processed at each server. Security breaches, occurring at any server, should be reported immediately. A complication is replicated web server and application server instances. Each server will be creating its own log that will have to be synchronised to allow integration.

As millions of entries could possibly be made in these logs and as they all have different formats, they are very difficult to integrate and analyse.

As a result, audit log integration and log analysis tools will become important mechanisms to additionally implement in the virtual web database environment.

The disparate servers of the virtual web database environment make this service more complicated to enforce, as stipulated in database security.

3.6 AVAILABILITY

Database security provides availability as a service to ensure that information is available to authorized users when they need it. Availability is the only service where the withholding of information pertains to both information and resources. [TRYF00]

Availability of the virtual web database environment

Both web server and application server availability can be improved with load balancing, where server instances are replicated to process a high volume of requests. This impacts on the security of the virtual web database environment, as all these server instances will need to have the same security configuration. In addition, the multiple web and application server instances must be able to maintain secure session state for a user.

The bottleneck of the virtual web database environment is the database server. Its availability cannot be as easily improved by replicating it in real terms, as in the case of the web server and application server. If various database server instances were used to process more requests, integrity of data would be compromised. Hardware solutions, or partitioning and replication of tables can improve database server availability as shown in table 1.

Backup and recovery procedures at each server must be implemented without fail.

Creating a highly available virtual web database environment has its problems. The infrastructure becomes more complex, and security assurance is impacted negatively.

3.7 MANAGEABILITY

Manageability is the ability to easily create and maintain the security mechanisms of the database as to aid in its confidence.

Manageability of the virtual web database environment

Manageability is a security service that can impact the level of assurance of security dramatically.

The enforcement of authentication, access control, audit and availability at each server will be the responsibility of the administrator. The number of administrators and security policies can quickly become unmanageable in the virtual web database environment. For instance, any change made to access control lists at the web server may impact the permissions assigned to roles at the application server and database server. These changes need to be propagated to servers by manually implementing them. This can result in an error-prone process.

Ensuring the security at each server of the virtual web database environment is therefore a time-consuming task. Very often, tools are available that assist in managing performance and availability, but real security issues such as integration of access control or audit logs are not addressed.

As poor administration and security management often lead to security breaches and end-user frustration, security management tools become important to ensure a secure virtual web database environment.

3.8 ASSURANCE

Assurance is the database security service that will determine the degree of confidence to which the security needs of the database are satisfied.

Assurance of the virtual web database environment

The large number of security breaches occurring at web servers is proof of the fact that their security is not assured. Creating a secure virtual web database environment implies that security is taken into account from the initial design phase. Thorough testing must be performed at all servers, at all times. Independent third parties, as well as vulnerability assessment tools can aid in this process.

The database server has an additional advantage in that its security can be assured. Ideally, the database server of the virtual web database environment should have been successfully evaluated with ITSEC or TCSEC criteria.

3.9 PHYSICAL SECURITY

Physical security ensures that the database is protected from unauthorized access, damage and interference. This is often overlooked, but should be the first step in securing all equipment and resources in the virtual web database environment.

3.10 NON-REPUDIATION

Non-repudiation is a security requirement that is not included in current state database security. Its requirement stems from the fact that virtual web database environments have to deal with unknown customers, who may be difficult to identify. Customers should not be able to claim at a later stage that they did not process transactions, if they in fact did.

The use of digital certificates will ensure non-repudiation. When a request to process a transaction is sent from the client to the web server, the digital certificate of the customer will accompany the request. The web server will verify the digital certificate and will send it to the application server as proof of the customer’s identity.

4. CONCLUSION

Even though database security services and mechanisms have not provided the perfect solution to securing databases, they could provide a basic framework for the security services to be provided by the virtual web database environment. This discussion has shown that virtual web database environments have security requirements that cannot fully be met by implementing database security services directly at each server. Careful integration of services, such as authentication and access control, needs to be done. As the implementation of a service has to be spread over all servers and is done by unreliable software, the security services are generally not implemented to the level stipulated by database security.

To ensure a secure virtual web database environment with current services and mechanisms takes a concerted effort. Each server must be actively configured, maintained and assessed to ensure some degree of confidence. The solution to this problem is therefore still evasive.

5. REFERENCES

[CAST95] Castano S., Fungini M., Martella G., Samarati P., Database Security, Addison-Wesley, 1995

[FERR99] Ferraiolo D., Barkley J., Kuhn D., A RBAC model and reference implementation within a corporate intranet, ACM transactions on information systems security, Vol.1, No 2, Feb 1999

[FRAN01] Franklin I., Protecting The Web Server And Applications, Computers & Security, 20 (2001) 31-35

[JOSH01] Joshi J.B.D., Aref W.G., Ghafoor A., Spafford E.H., Security models for web-based applications, Communications of the ACM, Feb 2001, Vol. 44, Issue 2, p38

[OSBO00] Osborn S., Sandu R.,Munawer Q., Configuring role-based access control to enforce mandatory and discretionary access control policies, ACM transactions in Information and Systems security, Vol. 3, No 2, Feb 2000

[PARK00] Park and Sandu, Secure cookies on the web, IEEEE Internet Computing, July-August 2000

[PFLE97] Pfleeger C.P., Security in computing, Prentice-Hall, 1997

[SAND94] Sandu R., Feinstein H., A three-tier architecture for role-based access control, Proc. Of the 17th NIST-NCSC National Computer Security Conference, Baltimore, Oct 1994, p138-149

[SSLI01] Introduction to SSL,Netscape, http://docs.iplanet.com/docs/manuals/security/sslin/index.htm [Feb 2001]

[TRYF00] Tryfonas T., Gritzalis D., Kokolakis S., A qualitative approach to information availability, 16th Annual working conference on Information Security, Aug 2000

[WISE01] Wiseman S., Database security: Retrospective and way forward, Information Security Technical Report, Vol. 6, No. 2, 2001, p 30-43