International Journal of Technology and Engineering System (IJTES)
Vol 6. No.1 – Jan-March 2014 Pp. 51-57
©gopalax Journals, Singapore
available at : www.ijcns.com
ISSN: 0976-1345

AUDIT DATABASE SECURITY POLICIES TO MITIGATE MALICIOUS ATTACKS USING REVERSE ENGINEERING PROCESS

Pratheepkumar Vajravel , Mr.P.Savaridassan
Information Security and Computer Forensic
SRM University
Kattankulathur, Chennai, Tamil Nadu, India
pradeepvajravel@gmail.com savaridassan.p@ktr.srmuniv.ac.in

ABSTRACT

Security policy describes rules placed on entities and actions in a system. Security policies and system mechanisms are not same since the system mechanisms are enforcing the policy. Neglecting security policies result in futile security controls because there is no awareness of the internal practices and company assets which users are attempting to secure. Defined security policies can help to make sure that systems are used in the intended manner; enable user understanding of their roles & responsibilities and control the legal liability. Although, we have suggested policies for effective security program to mitigate and retrieve vulnerabilities in software, since security policies are dynamic, it needs to be reviewed and changed regularly by security experts. In this context black hats effortlessly discover security vulnerabilities in software. As a first step this requires to discover the real security constraints being enforced by the database and to represent them at an appropriate abstraction level to enable their understanding and re-engineering by security experts. Although there are many techniques for Sybase database, in this project malicious attacks are defended by presenting a security meta-model using automated software based reverse engineering process that helps security consultant to visualize and evaluate security policies.

Index Terms—Database Security policies, Reverse engineering debug tools (ollydbg, installwatch pro2.5), Sybase DBMS, Audit tools, RBAC

1. INTRODUCTION

The concept of information security has been around for as long as there's been information worth securing. In recent years we saw continuous growth of interest in the information security; it essentially became a separate discipline, and many universities now offer degrees with concentration in Information Security. At the same time, as progress in technology and increased awareness helped to build more secure systems and processes, the nature of threats also evolves. New trends in information technology, like global development and outsourcing, growth of data processing powerhouses, software as a service (SAAS) also create new vulnerabilities. Several high-profile security breaches highlighted the need to proactively manage security, and often anticipate consequences of yet unknown vulnerabilities and threats. This changing landscape requires educators to continuously update their information security curricula, provide students with practical skills, and develop their ability for lifelong learning.

The classic CIA (Confidentiality, Integrity, and Availability) model has been the cornerstone of information security and numbers of systems have been built with this model in mind. Securing this information is therefore a critical concern. For this purpose of this policy is to secure the private sensitive information of organization, and to prevent the loss of information that is critical to the operation of the organization. Policy defines the technical controls and security configurations users and Information Technology administrators are required to execute in order to ensure the integrity and availability of the data environment. Security policies are the most crucial element of a security program. Without security policies there are no effective security controls because there is no awareness of the internal practices and company assets, users are attempting to control. Security policies clarify the security goals of an organization in relation to its

International Journal of Technology and Engineering System (IJTES)
Vol 6. No.1 – Jan-March 2014 Pp. 51-57
©gopalax Journals, Singapore
available at : www.ijcns.com
ISSN: 0976-1345

business processes, technical mechanisms and personnel behavior.

A good security policy can help to ensure that systems are utilized in the intended manner; enable user understanding of their roles & responsibilities; and control legal liability. Information resources are protected by the use of access control systems. Access control systems include both internal (i.e. passwords, encryption, access control lists, etc)

Information Systems Security Policy is the high level security policy that establishes security objectives and roles within the organization. Several database mechanisms may be needed in combination to execute a policy, e.g., triggers can be used to add fine-grained (fine-grained auditing to monitor specific database activities, such as actions on a database table) control on privileges, scattering the policy and increasing the complexity of the definition process. Furthermore, as security requirements are rarely static, This policy defines the security policies (i.e., confidentiality, integrity, availability) and supporting policies (identification & authentication, audit, accountability, non-repudiation), the classification of information, and the roles and responsibilities of those entrusted with establishing and enforcing the security policies (information owners, business owners, information custodians, users). The actual Information Security Policy if one exists, or more likely we can examine a body of work that is identified as Security Policy by those individuals reported as responsible for aspects of security.

Experts on information security policy, this serves a dual purpose

1. Get to the relevant documents faster, especially if the organization of documents is less than obvious.2. Get at implied policy, that is unwritten security policy, or other kinds of documentation and guidelines which is simply not labeled as such. Representing and processing the security policies at this logical level is much easier than a direct exploration of the database dictionary whose structure is unfortunately not standardized. Additionally, this logical model would also allow us to execute all analysis/evolution/refactoring operations on the security policies.

The goal of this project, first we provide a means to represent such logical models for security concerns in Sybase relational DBMSs. Secondly, we describe a reverse engineering approach that can automatically create this logical model out of an existing database. Reverse engineering, as a process aimed to represent running systems at higher abstraction level, has been proved useful in many domains, including database systems. However, these works have focused on the database structure and ignored the security aspects. a model-based representation given this enables to reuse in the security domain the large number of model-driven techniques and tools. Model manipulation techniques can then be applied to visualize, analyze, evolve, etc the model. Then, a forward engineering process could be launched in order to generate the new security policy implementation ready to import in the target database system.

2. ACCESS CONTROL META-MODEL

Sybase relational database-specific access control meta-model (RBAC), The SQL standards predefine a set of privileges and object types that can be used in the definition of a relational database. Meta-model provides a direct representation of these concepts to facilitate the understanding of the security policies linking the security elements with the schema objects constrained. Database schema refers to the physical data model. In other words, to the definition of the database structure, specifically of the tables, columns, indexes, views, data types, etc., typically expressed in SQL DDL (Data Definition Language), example as e.g. 'create table' statements. There is some potential for terminology clash around the term "schema": As a generic database concept, "schema" is the definition of the database structure as described above, regardless of which database user owns the object(s).

In Sybase, a schema is a central concept. It is a collection of database objects (tables, views, stored procedures, triggers, etc) owned by a particular user. Fig: 2[1]

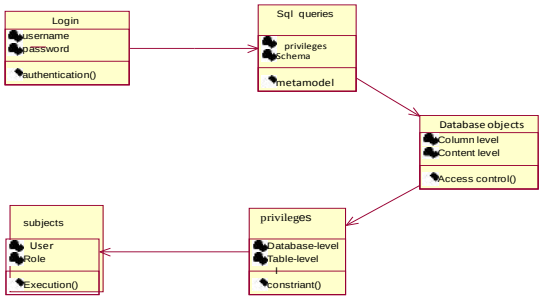

Figure: 1 sample meta-model Class diagram

International Journal of Technology and Engineering System (IJTES)
Vol 6. No.1 – Jan-March 2014 Pp. 51-57
©gopalax Journals, Singapore
available at : www.ijcns.com
ISSN: 0976-1345

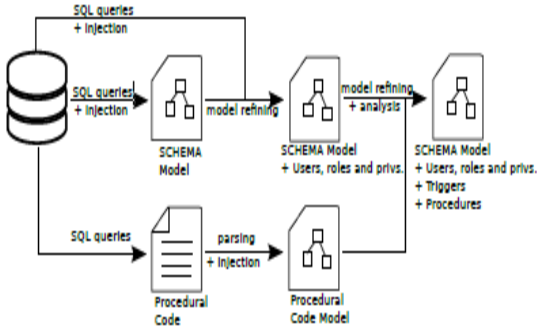

Figure: 2 Extraction process

Database objects: The main objects users can be granted privileges on are: tables, Columns, views, procedures and triggers. Each one of these elements is represented in the meta-model by a meta-class inheriting from the Schema Object meta-class. Views (meta-class View) can include derived columns. In fact, views can be used as a fine-grained access control mechanism. A view filters table rows and columns with respect to desired criteria so when a subject is granted privileges over a view, are actually being granted privileges over tables but with certain constraints. This actually represents column-level and content-based (row level) access control.

Privileges: We can divide the privileges that can be granted in DBMSs in five categories: Database-level privileges; for those privileges that imply creation of database objects including users and roles. Table-level privileges for those that implies table, columns and index access. Permission-delegation privileges to delegate permission administration to users and roles. Execute privileges for the executable elements (stored procedures and functions) and Session privileges. If needed, an extension of this meta-model (by inheriting from the Operation meta-class) could be provided to deal with vendor-specific privileges.

Subjects: Subjects executing actions on the DBMS are users and roles and they are, as such, represented in the meta-model with the corresponding meta-classes User and Role, Possibly part of role hierarchies. In the meta-model, the meta-class Role inherits from the Meta-class Subject which enables it to become grantees. The User meta-class also inherits from Subject what means that, privileges, in contrast to pure RBAC, can be granted to both users and roles.

Other elements and constraints: There are several other aspects that have to be taken into account. The execution of privileges may need to be constrained. In DBMSs this is normally achieved by using triggers and procedural code. The meta-model should permit the representation of the existence of such constraints.

The meta-class Constraint meets that purpose. Typically it points to a Trigger linked to the object and the operation on it that is constrained by the trigger. The Trigger meta-class also includes the attributes the trigger body and the condition and error message to be displayed when the trigger execution fails due to any exception. Another aspect to be considered is object ownership as it is the basis for permission delegation.

An association between objects and subjects records this relationship. Finally, global permission, i.e., permissions that are granted on all the elements of the same type is represented by allowing in the meta-model permissions to be granted on any object, including the meta-classes Schema and Database. A select permission granted on Schema or Database represents that the grantees can select all the tables and views contained in those objects.

A. Extracting general schema information

This first step populates the part of the model describing the structure of the database itself e.g., schemas, tables (and columns), views, procedures, etc. An injector (in our terminology, an injector is a software component that bridges technical spaces, in this case moving information from the database technical space to the modeling technical space) is needed. This injector connects to the database and queries the dictionary tables to retrieve all the necessary information. The selected objects are then inserted as model elements in the security model. In Oracle, our injector has to query tables like DBA TABLES, DBA TAB COLS and DBA ROLES. View extraction is more challenging since we need to parse the view definition to get the list of tables (or other views), columns and conditions the view selects.

The result of the parsing is a set of links between the view object and the other objects filtered by the view. The where clause will be copied as it is into the condition attribute of the view meta-class. Moreover, a view can contain derived columns. Derived columns are created as view columns in the model.

B. Extracting users, roles and privileges

As before, the injector queries the database dictionary for user and roles information and populates the model with the results. After this step, the general access control information of the

International Journal of Technology and Engineering System (IJTES)
Vol 6. No.1 – Jan-March 2014 Pp. 51-57
©gopalax Journals, Singapore
available at : www.ijcns.com
ISSN: 0976-1345

database, i.e., subjects, objects and permissions are already present in the security model.

C. Triggers information

Triggers: Complex security checks can be implemented by means of triggers that fire to prevent certain actions when performed in a certain context. However, triggers are used for a huge variety of tasks beyond security purposes. In our approach, all triggers are retrieved and parsed, then, they are analyzed with respect to a number of heuristic conditions in order to select which of them are implementing security checks and discard the rest.

a) The heuristic conditions analyze the following aspects

BEFORE STATEMENT triggers are executed before the statement is completed, enabling the possibility of evaluating security concerns (e.g., the possibility to make inserts in certain tables could be enabled only to working days).

Conversely, AFTER STATEMENT triggers are executed once the action of the statement is performed, so, when involved in security, they are normally used for logging purposes. Clearly, our focus should be in the BEFORE STATEMENT triggers. Trigger contents: Although the kind of the trigger is an important hint, it is not enough. We analyze the trigger actions and conditions to find operations that are likely to be used when performing security checks like system context information checks, user information checks, etc.

b) Sybase triggers syntax (sample)

Instead of trigger

```
create trigger [owner.] trigger_name
        on [owner.]view_name
        instead of {insert, update, delete}
        as SQL_statements
```

Instead of trigger with update clause

```
create trigger [owner.] trigger_name
        on [owner.]view_name
        instead of {insert, update, delete}
        as
                [if update (column_name)
                        [{and | or} update (column_name)]...]
                        SQL_statements
                [if update (column_name)
                        [{and | or} update (column_name)]...
                        SQL_statements] ...
```

c. Drop instead of trigger statement

The syntax for dropping the instead of trigger is the same as for the for triggers.

Parameters

1. trigger_name – is the name of the trigger, which must conform to the rules for identifiers and is unique in the database. To create another trigger of the same name, owned by a different user in the current database, specify the owner's name.
2. view_name – is the name of the view on which to create the trigger.

Insert, update, and delete –statements that can be included in any combination. Delete cannot be used with an **if** update clause.

SQL_statements – specify trigger conditions and actions. Trigger actions take effect when the user action (insert, update, delete) is attempted. If update – tests whether a specified column is included in the set list of an update statement, or is affected by an insert statement.

D. Access control mechanisms in DBMS

A security mechanism allows us to enforce a chosen security policy. Two main mechanisms at the DBMS level: 1.Discretionary access control, 2.Mandatory access control.

Discretionary Access Control

Based on the concept of access rights or privileges for objects (tables and views), and Mechanisms for giving users privileges (and revoking privileges). Creator of a table or a view automatically gets all privileges on it. DMBS keeps track of who subsequently gains and losses privileges, and ensures that only requests from users who have the necessary privileges (at the time the request is issued) are allowed.

Mandatory Access Control

Based on system-wide policies that cannot be changed by individual users, Each DB object is assigned a security class. Each subject (user or user program) is assigned a clearance for a security class.

Rules based on security classes and clearances govern who can read/write which objects. Most commercial systems do not support mandatory access control. Versions of some DBMSs do support it; used for specialized (e.g., military) applications.

3. REVERSE ENGINEER A MODEL FROM A DATABASE

Reverse engineering is the process of creating a data model from a database or a script. The modeling tool creates a graphical representation of the selected database objects and the relationships between the objects. This graphical representation can be a logical or a physical model. A database can be reverse engineered for the following reasons:

International Journal of Technology and Engineering System (IJTES)
Vol 6. No.1 – Jan-March 2014 Pp. 51-57
©gopalax Journals, Singapore
available at : www.ijcns.com
ISSN: 0976-1345

To understand how the objects are related to each other and then to build upon it. To demonstrate the database structure
After the reverse engineering process completes, you can perform the following tasks:
1. Add new database objects
2. Create the system documentation
3. Redesign the database structure to suit your requirements
Most of the information that you reverse engineer is explicitly defined in the physical schema. However, reverse engineering also derives information from the schema and incorporates it into the model.

A. Reverse Engineering process

Reverse engineering process is now frequently used on computer hardware and software based process. In this project using on software based reverse engineering process. Software reverse engineering involves reversing a program's machine code (the string of 0s and 1s that are sent to the logic processor) back into the source code that it was written in, using program language statements.

Software reverse engineering is done to retrieve the source code of a program because the source code was lost, to study how the program performs certain operations, to improve the performance of a program, to fix a bug (correct an error in the program when the source code is not available), to identify malicious content in a program such as a virus or to adapt a program written for use with one microprocessor for use with another. Reverse engineering for the purpose of copying or duplicating programs may constitute a copyright violation. In some cases, the licensed use of software specifically prohibits reverse engineering. Someone doing reverse engineering on software may use several tools to disassemble a program. One tool is a hexadecimal dumper, which prints or displays the binary numbers of a program in hexadecimal format.

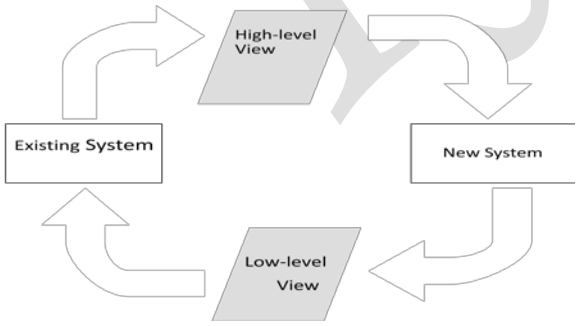

Figure: 3reverse engineering process architecture

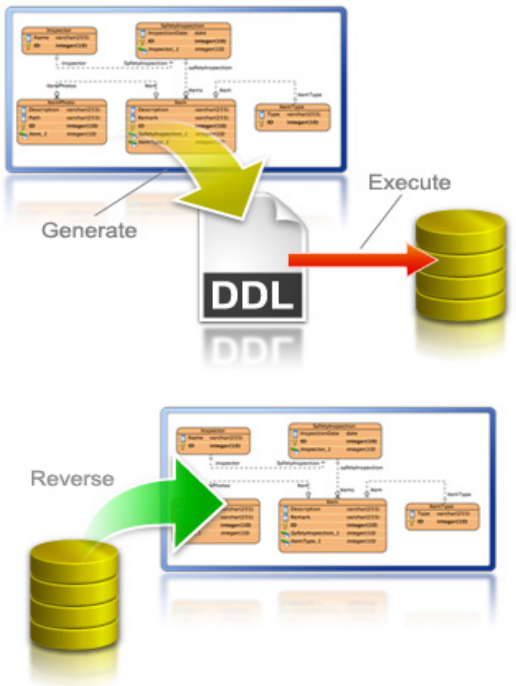

Fig:ure 4 forward & reverse engineering process in DBMS

B. Effective act of intend scheme

Sybase's adaptive server enterprise product includes a Policy-Based Access Control framework that provides means for protecting data down to the row level. Administrators can define security policies that are based on the value of individual data elements. Rules for access to resources have been established by the information/application owner or manager responsible for the resources. a person or group of persons need to access a database system, the individual or group must first apply for a user account. Individual users shall have unique logon IDs and passwords. An access control system shall identify each user and prevent unauthorized users from entering or using information resources. Security requirements for user identification include: 1.Each user shall be assigned a unique identifier, 2.Users shall be responsible for the use and misuse of their individual logon ID. The DBA will then create a new account id and password for the user if he/she deems there is a legitimate need to access the database. The user must log in to the DBMS by entering account id and password whenever database access is needed. The database system must also keep track of all operations on the database that are applied by a certain user throughout each login session. Roles or permission of user access for DBMS, user position of organization using the roles or permission granted in organization.

International Journal of Technology and Engineering System (IJTES)
Vol 6. No.1 – Jan-March 2014 Pp. 51-57
©gopalax Journals, Singapore
available at : www.ijcns.com
ISSN: 0976-1345

Roles can be created using the CREATE ROLE and DESTROY ROLE commands.

The GRANT and REVOKE commands discussed under Discretionary Access Control (DAC) can then be used to assign and revoke privileges from roles.

A credential is a set of properties concerning a user that are relevant for security purposes, For example, age, and position within an organization. To keep a record of all updates applied to the database and of the particular user who applied each update, we can modify system log, which includes an entry for each operation applied to the database that may be required for recovery from a transaction failure or system crash. A database audit consists of reviewing the log to examine all accesses and operations applied to the database during a certain time period. A database log that is used mainly for security purposes is sometimes called an audit trail. To establish the process for conducting, on a periodic basis, an operational review of system activity including, but not limited to, user accounts, system access, file access, security incidents, audit logs, and access reports.

As stealing data is pretty simple if you know how to do it and the bad guys are learning fast, they are investing and building attack tools while companies seem to be sleeping and giving away for free their data. One simple mistake can lead to database compromise. If you don't protect your databases sooner or later you will get hacked, this means a lot of money loses and in worst case running out of business. Perimeter defense is not enough.

The attack can be happened after taken audit report on security policies related, to comparison of both reports for before and after attack and then analyzing to find the weakness of security policies level to retrieve the vulnerabilities using software based reverse engineering process. it is defines Reverse engineering process machine code to source code (low level to high level abstract). Why use reverse engineering process was the original product design documentation has been lost or never existed in security administrator in organization. To covered vulnerabilities of program b/w database using meta-model to software based reverse engineering process using Role based access control policy Its analyze/evaluate the recreating program and retrieve the vulnerabilities of DBMS.

Reverse engineering process using INSTALLWATCH PRO25 and OLLYDBG TOOLS, they are used to analyze the security policies or registry on system database.

Example: Steps of INSTALLWATCH PRO25 TOOL

Step: 1 Open INSTALLWATCH PRO25 tool and click snapshot option, snapshot to analyze all files and registry of system database, for example install, delete and modify registry.

Although, taken snapshot report of all data's location current loop, this report taken before software installation or delete, check add or remove directories and initialized in system database.

Step: 2 Click install option of tool the dialogue box should be open, to select correct path of installation software and click enter. The software can be installing to the system, after software installed, to occupy again snapshot option to taken report.

Step: 3 the comparison of both snapshot reports are certain difference and figure out of changed on local system. This like to be sand box, find virus and worms are added in system check it and removed.

Short notes-- In my project, backend for Sybase DBMS and front end for java platform since java connect with Sybase database using JDBC or Hibernate driver. Java driver directly connected from DB schema, and use audit tool (diff tool) to analyze and monitoring security policies of database and to present a report for internal or external auditing methods and attempt to malicious attacks on Sybase database. Audit report to find vulnerabilities after attack, to comparison of both report to find weakness and strength of security policies and using automated software based reverse engineering process to retrieve the bug or virus for local system and protect the information. For example user privilege or permission is stored in meta-model table. Security manager to provide on read only permission for user. The organization provide to the user, unique access id and password.

4. DATABASE OBJECTS – COMMON ISSUES

A. Check for object and system permissions

Check views, stored procedures, tables, etc. permissions. Check file, folder, registry, etc. permissions. Changes on permissions could mean a compromise or misconfiguration.

B. Look for new database installations

Third party products can install database servers and new installed servers could be installed with blank or weak passwords, un-patched, misconfigured, etc. Detect new database installations and secure or remove them.

International Journal of Technology and Engineering System (IJTES)
Vol 6. No.1 – Jan-March 2014 Pp. 51-57
©gopalax Journals, Singapore
available at : www.ijcns.com
ISSN: 0976-1345

C. Search for users with DBA privileges

This helps to detect intrusions, elevation of privileges, etc.

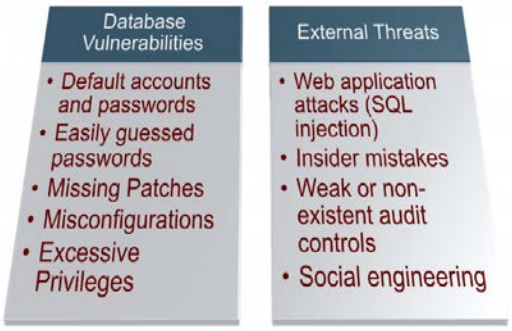

Figure: 5 Common Database Issues

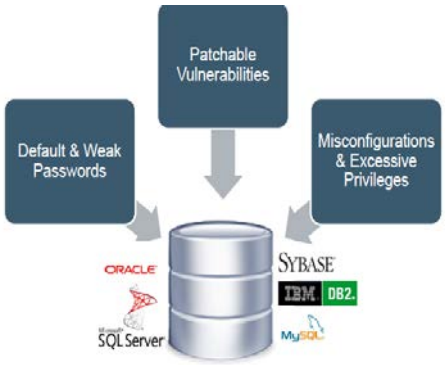

Figure: 6 Database Vulnerabilities

5. HOW TO PROTECT AGAINST ATTACKS

Set a good password policy: Use strong passwords or passphrases.

Keep up to date with security patches: Try to install patches as fast as you can. Database vulnerabilities are serious and try sometimes a database server can be easily compromised with just a simple query. Always test patches for some time on non-production databases

6. CONCLUSION

As information technology increases, the importance of security continues to become even much more critical. Sybase Adaptive Server Enterprise meets the need for a wide-range of security features for any organization, from the least to the most critical applications. Each of the security features discussed in this paper can be utilized within ASE independently or in combination. Discretionary Access Controls feature is also optional and at the system administrator's discretion, other system users can be given permissions to perform system type functions. The Policy-Based Access Control feature is optional and will allow the security control to operate at yet a more granular level for protecting data. Access Rules are optional and work in conjunction with the previous feature Policy-Based Access Control, therefore allowing more robust to security controls. These features will allow any organization to provide protection to the greatest. With the ease of use, the features will allow for one to implement security for its system/data in a minimum amount of time with little training, thus providing required security in a reasonable time.

REFERENCES

[1] SalvadorMartınez,(2013),Reverseofdatabasesecuritypolicies,Conf:AvailablePDF:http://docatlanmod.emn.fr/DatabaseRE/DEXAShortPaper2013.pdf

[2] Implementing Database Security and Auditing by Ron Ben Natan Chapter 12 & 13.

[3] http://www.healthit.gov/sites/default/files/tools/info_security_policy_template_v1_0.docx

[4] http://www.hinfonet.org/sites/default/files/resources/Information%20security%20policy%20templates.doc

[5] J.-M. Petit, J. Kouloumdjian, J.-F. Boulicaut, and F. Toumani. **Using queries to improve database reverse engineering.** ER '94, pages 369–386. Springer, 1994.

[6] R. H. L. Chiang, T. M. Barron, and V. C. Storey. Reverse engineering of relational databases: extraction of an EER model from a relational database. Data Knowl. Eng., 12:107–142,March 1994.