called “Security” to describing the virtual private database features. The example is a human resources application. This also contains lots of detail and suggestions for ease of management.

TIPS, TRICKS, CONSIDERATIONS AND CAVEATS

The step-by-step articles all try to demonstrate the process in simple terms. Most allude to possible complications in the world beyond the textbook examples. The following is a compilation of suggestions, some explicit and some suggested between the lines.

- “...the ability to create a security context is a separate system privilege; only suitably-privileged users are able to create a context.”[20] As always, good security practice requires that some thought be given to which groups of users (developers, DBAs, schema owners) should be given the privileges needed to create a security policy.

- “...it's a good idea to have the policy function owned by a system security officer, to prevent a developer or user from inadvertently or maliciously dropping a policy from a table.”[21] That is, the security policy should be created by a schema / user other than the schema which owns the tables and views protected by the policy. There should be restricted access.

- Security policies need to take into account parent-child relationships between tables. Applications may force users to retrieve information from the parent or master table first before selecting from the child or detail tables.[22] Again, the policy needs to be written on the assumption that the user is bypassing the restrictions built into the application.

- “If the function returns a zero length predicate, then it is interpreted as no restriction being applied to the current user for the policy.”[23] The principle of least privilege would suggest that security functions should be written to provide explicit access to users meeting certain conditions, and to provide no access to the users who fail to meet any of those conditions. Accordingly, a ‘none of the above' option should build a predicate that will deliberately fail to retrieve any rows from the table. The code examples in the step-by-step articles show several ways of accomplishing this.

- “SYS user is not restricted by any security policy.”[24] The implication here is that other users, including SYSTEM, are. This may affect exports performed by SYSTEM. The schema which owns the tables and views may also be restricted by the security policy. This may require some adjustment in the assumptions made by developers who are used to the schema owner automatically having full rights to every row of every table. There is an EXEMPT ACCESS POLICY system privilege, but as with any system privilege, it should be used judiciously.

- Security policies need to account for super users’ activities. Many applications

require batch processing. Some user, frequently the application owner, must be authorized by the security policy to perform the necessary updates on all rows of the table.

- It is possible to construct a security policy function that causes an infinite loop.[25] Suppose information from the employee table is used in determining which rows the user can access, and that the security policy function which protects the employee table needs to query the employee table to find that information. The suggested remedy is grant access to the table only to the security manager user which creates the policy, and to have all other users access a view of the table. The security policy functions for the view would select the attributes which determine access from the original table.

- Errors in the application context invoked by a database trigger on logon could prevent all users from connecting to the database.[26] As security policies do not apply to the user SYS, SYS is able to come to the rescue.

- It is possible to create a context using a schema and package that does not exist.[27] This can cause confusion and name conflicts, particularly in the development environment, unless there is a periodic procedure to find and remove the contexts that refer to phantom schemas and packages.

ORACLE 9I ENHANCEMENTS

The basic VPD available in Oracle 8i applies a policy for a given user, regardless of which application the user is using to access the database. Oracle9i offers some enhancements, particularly for the n-tier application model.

Partitioned FGAC allows for different policies to be applied to the same tables and views for the same user, depending on which application the user is using. This allows applications to use 3rd party software tools, such as reporting tools. It simplifies development because the developers of different applications using the same data don't have to coordinate their efforts to define a single policy that addresses all possibilities. The server detects which software is accessing the data during a given session, and activates the appropriate policy. Better yet, when in doubt, the server concatenates all policies so that when no specific policy is in force, all policies are in force.[28]

The **global application context** allows an application server program to identify an end-user who is connecting to the application as if that end-user were connecting directly to the database. This is especially good for auditing activity at the database level. Otherwise, the application has to duplicate functionality that is available in the database, which increases cost of development and maintenance.[29]

Oracle9i Label Security is an additional cost option that is useful when the characteristics of the data to be protected are not sufficient to describe the data's sensitivity. "Oracle Label Security uses the application context functionality of the VPD

product"[30] , but where FGAC uses a dynamic predicate, OLS uses labels applied to both the data and to the users to control access.[31]

"Oracle Label Security is based on data element labeling concepts used by government and defense organizations to protect sensitive information and provide data separation."[32] One of the elements of the label is the sensitivity of the data. The other elements of the labeling scheme are 'compartment' and 'group'. These are likely to be the attributes similar to ones a FGAC security policy would use. When individual rows in a given compartment and group have different sensitivity levels, these compartment and group characteristics are not enough to identify which users should have access to the rows.

Spy movies have made us all familiar with ratings like 'classified' and 'top secret', and we are also comfortable with the idea that a person with 'top secret' clearance would be able to access information at all sensitivity ratings up to and including 'top secret'. The sensitivity hierarchy, in combination with policies defined by the application context, provides an extra layer of protection.

OLS includes an EXEMPT ACCESS POLICY system privilege, which provides a way to exclude a user from the security restrictions.[33]

Oracle-Base provides a simple step by step guide to setting up an Oracle Label Security policy.[34]

FINE GRAIN AUDITING

One of the insider problems mentioned at the start of this paper was the super user who uses his / her privileges inappropriately. The principle of least privilege and separation of duties help prevent problems in this area, but there also needs to be a way to detect whether there have been violations of policy. Auditing is a tool that can help, but it often generates so much data that it is more useful for proving a suspected problem than flagging one for investigation. Loney recommends "enabling auditing for certain key events, reporting on those events frequently, and truncating the SYS.AUD$ table regularly" as one of his six tips for protecting an Oracle database.[35]

Fine grained auditing is a tool that can help identify key events, particularly when there are concerns about the privacy of the data. "Oracle FGA provides facility to identify abuse of legitimate user privileges and possible intruders while reducing the volume of unnecessarily audit data that would normally be generated by turning on full-blown auditing." [36]

The concept is similar to fine grain access control. The fine grain auditing policy defines the schema and object to be protected, a condition that will trigger an audit entry, and the information to be captured for the audit entry. FGA identifies conditions that would classify the row as sensitive, and then reports specific actions against the sensitive rows. For example, a law enforcement agency might want to be alerted whenever

someone attempts to look up the license plate of one of its undercover vehicles.

SUMMARY

"The compromise between enabling appropriate database access and maintaining tight security against unauthorized access is the enduring dilemma of the database administrator". [37]

Oracle's Virtual Private Database features and related products give developers and database administrators powerful tools to protect the confidentiality and integrity of the organization's most valuable data.

REFERENCES

Ault, Michael R., TUSC. "Managing Row Level Security in Oracle 8i". June 2001.
URL: http://www.quest-pipelines.com/newsletter-v2/rls.htm

Browder, Kristy and Mary Ann Davison. "The Virtual Private Database in Oracle9i R2 - Understanding Oracle 9i Security for Service Providers - An Oracle White Paper". January 2002.
URL: http://otn.oracle.com/deploy/security/oracle9iR2/pdf/VPD9ir2twp.pdf

Davison, Mary Ann. "Creating Virtual Private Databases with Oracle8i". Oracle Magazine. July 1999
URL: http://www.oracle.com/oramag/oracle/99-Jul/49sec.html

Davison, Mary Ann. "The Need for Granular Access Control". Secure Business Quarterly. Volume Two, Issue Two, Second Quarter 2002.
URL: http://www.sbq.com/sbq/app_security/sbq_app_granular_access.pdf

Feuerstein, Steven. Guide to Oracle8i Features. O'Reilly & Associates, Inc, 1999. p.153-170

Loney, Kevin. "Protecting Your Database", Oracle Magazine, May 2000.
http://www.oracle.com/oramag/oracle/00-May/index.html?o30sec.html

Oracle-Base, "Oracle Label Security (OLS)",
URL: http://www.oracle-base.com/Articles/9i/OracleLabelSecurity9i.asp

Oracle-Base, "Virtual Private Databases (VPD)", URL: http://www.oracle-base.com/Articles/8i/VirtualPrivateDatabases(VPD).asp#SecurityPolicies

Scherer, Douglas, William Gaynor, Jr., Arlene Valentinsen, Xerxes Cursetjee. Oracle8i Tips & Techniques. Osborne / McGraw-Hill, 2000. p.193-217

Sherman, Roby “Internet Security With Oracle Row-Level Security”
URL: http://www.interealm.com/roby/technotes/8i-rls.html

Sherman, Roby “Implementing Data-Level Monitoring With Oracle Fine-Grained Auditing” .
URL: http://www.interealm.com/technotes/roby/fga.html

Smith, Howard. “Hack Proofing Oracle”.
URL: http://otn.oracle.com/deploy/security/pdf/oow00/orahack.pdf
(Note: Oracle Technology Network (otn) requires a user account and password)

Ziola, Brad. “Label Based Access Control vs. Fine-Grained Access Control for Implementing a Virtual Private Database”. March 2002.
URL: http://www.managedventures.com/images/OLSFGAC.pdf

ENDNOTES

1. Davison, “The Need for Granular Access Control”, p. 2
2. Browder, p. 1
3. Smith, p. 6-7
4. Davison, “Creating Virtual Private Databases with Oracle8i”, p. 2
5. Davison, “Creating Virtual Private Databases with Oracle 8i” , p. 2
6. Scherer, p. 197
7. Browder, p. 6
8. Scherer, p. 197
9. Scherer, p. 210
10. Davison, “Creating Virtual Private Databases with Oracle 8i” (Part 2)
11. Browder, p. 5
12. Browder, p. 5
13. Oracle-Base, “Virtual Private Databases (VPD)”
14. Smith, p. 5
15. Davison, Mary Ann “Creating Virtual Private Databases with Oracle 8i”
16. Sherman, Roby “Internet Security With Oracle Row-Level Security”

17. Ault

18. Feuerstein, p. 153-170

19. Scherer, p. 193-211

20. Browder, p. 7

21. Davison, “Creating Virtual Private Databases with Oracle 8i”

22. Davison, “Creating Virtual Private Databases with Oracle 8i”

23. Ault, p. 6

24. Ault, p. 6

25. Scherer, p. 203

26. Feuerstein, p. 169

27. Scherer, p. 212

28. Browder, p. 8

29. Browder, p. 9

30. Ziola, p. 1

31. Ziola, p. 3

32. Ziola, p. 1

33. Ziola, p. 2

34. Oracle-Base, “Oracle Label Security (OLS)”

35. Loney

36. Sherman, “Implementing Data-Level Monitoring With Oracle Fine-Grained Auditing”

37. Ziola, p. 1