An Appraisal to Overhaul Database Security Configurations

Vamsi Krishna Myalapalli

Department of Computer Science & Engineering, JNTUK Vizianagaram, Andhra Pradesh, India.

***Abstract*-** Security engineering isa vital aspect in the Database Management Systems, since databaseis being targeted incessantly for vulnerability to invade sensitive data. Cherishing the database from security threats necessitates maintaining secure practices. The proposed model is intended to serve as benchmark for the DBAs,for overhauling the database configurations to withstand against threat waves.This paper enunciates different security configurations, independent of the proprietary DBMS engines that can serve as anassessment tool.

***Index Terms*-** Database, configuration, Security system, security policy.

I. INTRODUCTION

It is reckoned that amount of information is rapidly increasing and engendering magnitude of database to be increased further. Augmented information triggers vulnerabilities to be exploited further to invade database.

The following figure depicts the activity of genuine user vs. Intruderin interacting with database. The vulnerability can be an inefficient software design, which could be exploited by an intruder, (since he/she can't triumph over authentication system). Hence secure coding should be mandatory in practice, during application or interface development.

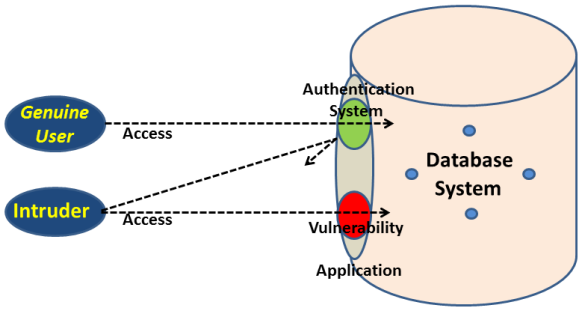

Figure1: Invading Database with exploited vulnerability

The security policy would comprise entities such as Authentication, Discretionary Access Control, Mandatory Access Control, and Administration. On the other hand comprehensive secure Database comprisesecure relational database, secure object database, secure distributed database and secure concurrency control.

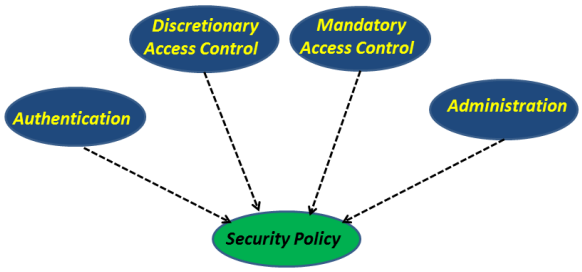

Figure2: Security policy and its entities

This paper lists all the security configurations that the DBA or the security expert should glean at. Rest of the paper is organized as follows. Section 2 deals with the associated background and related work. Section 3 deals with proposed benchmark. Section 4 demonstrates experimental setup. Section 5 is concerned with Comparative analysis and finally section 6 concludes the paper.

II. BACKGROUND AND RELATED WORK

Defense information systems Agency specified set of criteria with security implementation guide. Red Team strategy [11] entails a group of engineers, and they strive to hack the computer systems to quest vulnerabilities. Department of Defense forwarded Trusted System Evaluation Criteria which is referred as Orange book.

DBMS THREATS:

In contemporary practice DBMS is prone to following odious threats.

SQL Injection: It is the prevailing case where applications use Dynamic SQL. Dynamic SQL is the SQL compiled on fly when the application is running.

SQL Worms: They can be prevented by applying service packs and patches.

Genuine escalated privilege: The case where misfeasor misuses his privileges.

Authentication weakness: system vulnerabilities are exploited for authentication.

*Side channel exposure:*The case in which files are exposed physically. Ex: Stolen backup files.

Audit Trail feebleness: This case designates that sound audit plan is not implemented.

III. DATABASE ACCESS POINTS

Database security encompasses the access points User, Application, Network, Operating System, DBMS, Data files or

Data. The security risk increases as the hierarchy drives up. Correspondingly the number of access points to database reduces as the hierarchy drives down.The gap between each layer indicates missing security and thus engendering vulnerability. This is depicted in following portrait.

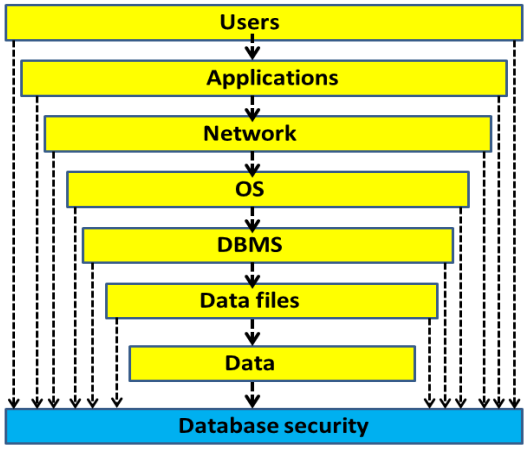

Figure3: Access points of Database security

IV. PROPOSED SYSTEM

The proposed benchmark is ramified into following categories Technological, managerial, Cultural practices each possessing their own criticality at different circumstances.

3.1. Technological: The technological facet deals with selecting, installing, configuring and monitoring DBM Sengine engendering effective access, security and data storage. In abroad manner Technological aspect is scattered into Environment, Installation setup, System level configuration, and Application level configuration and usage.

Environment:

1. Thwarting direct DBMS login onto user account
2. Granting only required resources for the user.
3. Deploying particular user to run daemons of DBMS.
4. Obliterating all unauthorized network protocols.
5. Making sure that unauthorized people can't physical access DBMS machine.
6. Implementing firewalls for network border and on machine.
7. Preventing machines that run critical network procedures such as authentication, naming etc.
8. Dedicating a separate machine for the Database.

Installation Setup:

9. Assign password to listener & lockdown access to it, according to database security policy.
10. Configure execute or read permissions only to genuine users.
11. Proper configuration of file owners.
12. Dedicating separate network sections for development and production servers.
13. Production server shouldn't be accessed by any of the developer.
14. Development and production servers should split physically.
15. Default SSL certificate usage must be deterred.
16. Mutating TCP and UDP ports.
17. Nomenclature for remote identification should be different.
18. Obliterate example databases.
19. Isolating OS files from DBMS software.
20. Data partitions can be read/write only by DBMS user.
21. Making partition for Database data.
22. Log partitions can be read or write only by DBA.
23. Log information is written onto a separate partition.
24. Change or remove default user names and passwords.

System Level Configuration:

25. Don't leak network data to allow connection attempts.
26. Use valid node checking to protect Database.
27. Generate strong cryptographic hashes of executables and other static files on host.
28. Self-signed certificated should not be used.
29. Remote authentications should be deterred.
30. Don't dump shared memory.
31. Restrict access to host as much as possible. Monitor everything that user does, if host access is granted.
32. Create processes and policies to ensure secure data going forward.
33. Limit access to executables.
34. Error and Trace logs should only be accessible to DBAs.
35. Turn off access to OS resources.
36. Access to OS level commands and information should be exterminated.
37. Unwanted access to additional functionalities and libraries should be dropped.
38. Don't allow network operations such as socket opening etc.
39. Read or write operations from DBMS on file system should be denied.
40. Preventing side channel leak of data from backup and configuration files.
41. Preventing side channel leak of data from trace files or core dump.
42. Preventing side channel leak of data from system variables.
43. Preventing side channel leak of data from configuration files.
44. Replicating the audit files, ensuring that they can't be lost.
45. Replicating the log files, ensuring that they can't be lost.
46. Implement proper logging system.
47. Access to sensitive data must be audited.
48. Enforcing sound encryption in storing passwords.
49. Passwords should be changed frequently.
50. Enforce password expiry after a predefined time.
51. Enforce locking after specified number of login failures.

52. Implement sound password policy.
53. Thwarting authentication based on host.
54. Making sure that remote parameters are not used for authentication.
55. Terminating idle connections.
56. Implementing sound user level authentication.
57. Configure access permissions to use IPs instead of host names, in order to prevent DNS spoofing.
58. Encouraging the use of server side certificates.
59. Strong encryption algorithm usage for remote communication.
60. Random ports shouldn't be used for connections of client.

Application Level Configuration and Usage:

61. Remove default schemas that are not required.
62. Review privileges and access to key data.
63. Lock and expire default user accounts.
64. Database schema creation SQL files should not be maintained in database server.
65. Encrypt the columns for sensitive data at table level. (Transparent Data Encryption)[9].
66. Turn off XML Databases.
67. Implement Single Sign On (SSO). (Helps to detect real user behind connection).
68. Encrypt or protect the code of application.
69. Turn off external procedures.
70. Restrict access to individual records within tables.
71. High privileged roles must be set a password.
72. Default connect role should be limited in order to restrict access to external procedures and Database links.
73. Securing batch jobs.
74. Roles or groups should be granted privileges instead to users.
75. System properties or configurations shouldn't be changed by any user.
76. Right assignments should not be delegated.
77. Direct table access should be deterred against stored procedures.
78. Should not use ANY or ALL during assigning rights.
79. Enable database link login encryption, in order to cherish privacy credentials.
80. Enforce Attribute-Based Encryption forFine-Grained Access Control of Encrypted Data.
81. Enforce the policy of least privilege in assigning rights.
82. Deny user quota(s) on system area.
83. User rights on system tables should be removed.
84. Restrict default roles.
85. Restricting public privileges as much as possible.
86. Alert or block abnormal access requests, in real time.
87. If a procedure of a restricted package is to be given access to a user, then create aprocedure and grant access on this procedure, to access procedure from restricted package, instead of granting rights on the whole restricted package.
88. Deny general application users to connectdirectly to the schema that owns the objects.

3.2. Managerial: The managerial aspect is concerned withadministrative functions. Efficient management leads to high quality information system.

89. Enforcing time to time backups.
90. Testing the backups.
91. Updating DBMS software.
92. Create a monitoring process at both system and database levels- this can be a policy, set of scripts, or a commercial tool.
93. Enable Data Dictionary protection.
94. Apply security patches, to reduce window of exposure.
95. Create Virtual Private Databases (VPD), which ensures abstraction [9].
96. Build your best practice auditing which discovers suspicious activity, access to sensitive data & access to auditing tool.
97. Review error logs regularly.
98. Encrypting database online and offline backups, thwarting backup data exposure or side channel exposure.
99. Configuration files and DBMS application should be examined for modifications.
100. If internal auditing is not possible (for some DBMS engines like MySQL) achieve this by creating triggers.
101. Periodically checking password hashes.

Compliance:

102. Measure and track configuration compliance over time.
103. Enforce accessibility to archived data in real time, if necessary to comply with an audit.

3.3. Cultural: The Cultural aspect deals with corporate resistance to adverse situations and change.

104. System administrator and Database administrator should not be the same individual.
105. Hire hackers to try to break into our systems, so that we can overhaul the vulnerabilities.
106. Production data should not be copied for testing purposes or sharing it with business partners.
107. Enforce separation of duties at IT level.

The following table depicts the criticality of each perspective specified above.The practice numbers which are quantified in the table are considered as very important, when compared to other secure practices.

Importance	Practice #
Critical	5, 6, 14, 59, 75, 100
Vital	3, 7, 8, 23, 24, 48, 52, 83, 89, 91, 98, 104
Suggested	1,10,13,15,20,22,34,36,45, 47,50,51,54,55,56,58,71,81

Table: Criticality of each practice

V. EXPERIMENTAL SETUP

Since it is not possible to demonstrate each practice, the following critical practice is explained via queries that serve as an exemplary.

#100. Create table *login*(name varchar2(30), userid number(3), intime timestamp);
Create table *logout*(name varchar2(30), userid number(3),outtime timestamp);
Create trigger *logintrigger*
After login to schema
Begin
insert into *login* values
(user,uid,systimestamp);
end;
Create trigger *logouttrigger*
before logoff on schema
begin
insert into *logout*
values(user,uid,systimestamp);
end;

Now the login and logout tables would entail login and logout details respectively for all users.

VI. COMPARATIVE STUDY

The benchmark assists the DBA, in each and every perspective, with the quantified criticality for each practice. The trust worthiness of the entire system can be determined by examining the system against the proposed benchmark, there by reckoning anticipated security impact. The database environment exhibits more secure form, since all the access points are secured against the threats.

VII. CLOSING COMMENTS

This paper proposed an appraisal to overhaul database security configurations and gazes at uplifting existing security engineering methodologies. In order to tackle the database security threat waves, more secure and innovative challenges should be instigated.

FUTURE WORK: I extend this work by bringing out more appraisals and methodologies by testing and discussing with several DBAs and experts.

REFERENCES

[1] Database and Applications Security. Bhavani Thuraisingham. Auerbach Publications.

[2] A Trust-Based Benchmark for DBMS Configurations.AfonsoAraújoNeto, Marco Vieira. 2009 15th IEEE Pacific Rim International Symposium on Dependable Computing

[3] Benchmarking Untrustworthiness in DBMS Configurations.AfonsoAraújo Neto, Marco Vieira. 2009 Fourth Latin-American Symposium on Dependable Computing. IEEE

[4] Appraisals Based on Security Best Practices for Software Configurations.AfonsoAraújoNeto, Marco Vieira. 2009 Fourth Latin-American Symposium on Dependable Computing. IEEE

[5] Hook-based Configurable Security for Pervasive Database.XinguoXu, JianfengQiao, Jiqin Du. IEEE

[6]

[7] Advanced SQL Database Programmers Handbook.Donald K. Burleson, Joe Celko, John Paul Cook, Peter Gulutzan. Rampant Techpress.

[8] Applied Oracle Security: Developing Secure Database and Middleware Environments.David C. Knox, Scott G. Gaetjen, Hamza Jahangir, Tyler Muth, Patrick Sack, Richard Wark, Bryan Wise. Oracle Press.

[9] Effective Oracle Database 10g Security by Design. David Knox. Oracle Press.

[10] Sandia National Laboratories, "Information Operations Red Team and Assessments™".

[11] Department of Defense, 1985. Trusted Computer System Evaluation Criteria.

[12] Database Systems. Design Implementation and Management 7th edition. Peter Rob, Carlos Coronel. Cengage Learning

[13] en.wikipedia.org/Database_security referred on 27th December, 2013.

AUTHORS

First Author – Mr. Vamsi Krishna.M is a student of JNTUK University College of Engineering, Vizianagaram, Andhra Pradesh, India. He is pursuing Bachelor of technology in Computer Science and Engineering.

He is Oracle Certified DBA Professional 11g, Database Security Implementation Specialist, SQL Expert, & JAVA SE7 Programmer. He is fervor towards database security and associated research. His other areas of interests entail SQL & Database Tuning, Programming etc. He can be reached at vamsikrishna.vasu@gmail.com