**Introduction**

Oracle is a relational database available in all major platforms (Windows, Unix…etc.). Despite its popularity due to great features and the integrated set of development tools, Oracle has its drawbacks that should be addressed, according to the context. The following report includes the types of security threats and failures that might occur within the Oracle database environment and an evaluation of various security measures, backup and recovery strategies that could be implemented in the company database.

Before diving into the problems and solutions related to database security and failures, it is better to understand the context of the problem, the environment of the company. The structure of the company is such that it has staff belong to different areas. Each area has a manager who needs information on the performance of the staff. The office administrator manages the performance information, staff details and general information about the staff. Helpdesk operators help customers and allocate problem numbers which they can use to backtrack or check progress. If the customer complaints are complex enough to escalate, they will be allocated to specialists in the problem area. Also, there is an IT support staff to keep the software and hardware up-to-date. A centralized database holds the helpdesk department details of the company.

**Types of security threats and failures that might occur within the Oracle database environment.**

Considering the nature of the functionalities in the company, a foremost security threat for the database is data theft and eavesdropping. A data theft exposes confidential and protected information. It could be an illegal data transfer or storage of information that is financially, socially, or personally confidential. Data theft or eavesdropping is possible to occur in a point of sales system remotely attacked by hackers. The consideration must aim at internal staff as well. Since internal attackers know the worth of data, it can be stolen and misused. It can be in the form of physical theft in a case where the devices left uncovered or unlocked. Crimeware such as malware, ransomware, SQL injection, and phishing attacks also lead to data theft and eavesdropping of data. Therefore, it needs to take relevant measures regarding devices used by helpdesk operators at the front line since they handle caller data extensively. USB drives are considered the easiest of such devices. Portable hard drives, devices using memory cards, personal digital assistants (PDA) are also among the devices that can be used in attacks. Data can be stolen or eavesdrop along the network communication lines. Therefore, the security authorities must be vigilant in the use of email, the printing of electronic data, and remote sharing of data. Through insecure communication lines and inside vulnerable servers, data theft is common to find.

Data can be at two states. Data at rest or data in motion. At both stages, data can be subjected to tampering. Data tampering is modifying, destroying, or manipulating data, decreasing its integrity at a large scale. The communication protocols between oracle database servers and clients in the company, need to be examined to understand how to protect the data in transit and eliminate unnecessary protocols that could threaten the data security. Monitoring network traffic, database logs, other system logs, and monitoring memory consumption of the system is needed to carried out to locate abnormal behaviors of the database. An attacker can exploit three areas to break into a system (Tripathi and Meshram, 2012). Weaknesses of configurations of a system, vulnerabilities of software, or the trust that has been given to inside staff personnel. There are various vulnerabilities in software that oracle has suffered most from at different points in times (Litchfield, 2007). Buffer Overflow attacks, PL/SQL Injection, trigger abuse, attacks via oracle application server are some of which need to be highlighted in the company context.

Buffer overflow attacks occur when too much data is inserted into the database which causes a buffer overflow and those data overwrites and makes a situation for an attacker to manipulate the other existing data. Many Oracle stored procedures have buffer overflow bugs. Oracle has released patches that need to be applied to eliminate these vulnerabilities. Depending on the oracle version being used, it is needed to apply the relevant patches to the DB. SQL injection is another common vulnerability. A SQL injection attacker can trick the SQL engine into executing unintended commands by supplying specially crafted string input, and then gain unauthorized access to the database to view or manipulate restricted data. PL/SQL injection can occur through SELECT statements, DML (UPDATE, DELETE, INSERT) statements, or anonymous PL/SQL blocks in procedures. Protecting DB from SQL injection is a part of the user privileges management plan. There is much that can be done by the programmer’s side such as, validating inputs to queries, using parameterized queries, avoiding string-building techniques to generate SQL, and using a web application firewall in front of the webserver to prevent SQL injection attacks. The technical managers must pay attention to these version updates, programming best practices and emphasize the engineers to implement them cautiously.

Triggers too are vulnerable to PL/SQL injections. A trigger program fires when a specific event occurs. Several triggers have been found to contain security weaknesses. Sometimes default triggers in Oracle can be abused to gain elevated privileges. In Oracle, the statements in the trigger body operate under the privilege domain of the trigger's owner, not the privilege domain of the user issuing the triggering statement. Therefore, using these privileges, anyone can get the trigger to execute an arbitrary function. Oracle has released patches for these trigger vulnerabilities as well.

The exploitation of configuration vulnerabilities was common before Oracle 10g as it had many default user accounts with default passwords. After 10g the passwords selected by DBAs were obfuscated but the files recording the passwords were known and accessible. Falsifying user identities, password-related threats, TNS poisoning, identity thefts, stealing data using rootkit and backdoor are some other threats which all related to authentication and authorization of access. Falsifying user identities happens mostly in distributed environments where clients and servers connect through networks. Hackers/hijackers interrupt the communication networks and access servers with false identities. TNS poisoning is a man-in-the-middle attack where the attackers hijack the legit user sessions and routes the connection to attackers’ servers. Mostly oracle is installed with default settings and ports so it is easier for attackers to locate the servers. After locating the servers, with few commands it can retrieve information about the Oracle database listener. It can register a malicious database service as the same service name in Oracle and can route the users to attackers' servers. From Oracle 12c version onwards this vulnerability was fixed with some configurations. With the IT team keeping the software and hardware versions up-to-date we can have faith in protecting the system from such attacks but streamlining the duties of IT staff will keep the updating procedure more reliable.

A rootkit is a collection of tools and software that can inject a malicious program into a system. The existence of a rootkit is extremely hard to detect. It can either be injected with a specific procedure or can come into a system piggybacking software a user trust. A rootkit provides privileged access to a system and allows an attacker to manipulate a system and monitor or change the files or procedures of a system without the knowledge of the user. Therefore, emails, links, files, and attachments, installing software, accepting license agreements should be done cautiously. A backdoor is a way to get inside a system without proper authentication. It can come as a part of software or through vulnerabilities of dynamic references of scripts. Once installed, the backdoor can hardly be detected. An attacker can manipulate an Oracle database by combining the rootkit and backdoor. A fine example is to use a rootkit to inject a backdoor in Oracle Listener so that the attacker can get access to the shell command remotely.

The best way to prevent rootkit and backdoor attacks is to use a cryptographically strong fingerprint technology like the MD5 algorithm which can check the integrity of the system files. In addition to this, every system must be updated, firewalled, and protected to avoid rootkit and backdoors. Monitoring network traffic, avoiding downloading cracked software, using next-gen antivirus programs, regularly scanning the system for malware or any suspicious programs, and most importantly learning and teaching the database and software engineers safe ways to access the data while working is much important in preventing these security threats to the database.

The importance of preventing security failures and defending against security threats has become a crucial consideration in a database administrator role. The advancements of technology have been able to produce sophisticated security mechanisms today. Yet the intruders have also developed anti-cybersecurity mechanisms to fight against. In a fast-paced business environment, a simple breach can have ripple effects that would drastically impact a company. A few such effects would be, damaged company reputation, lack of trust from clients, disturbances to business continuity and financial stability and, loss of privacy.

The following section describes the recovery strategies and backup mechanisms that can be practiced and adhered to, to ensure the data security of the company.

**Evaluation of security measures, backup, and recovery strategies that could be implemented to ensure the data security, availability, and integrity in the given company.**

The basic security standards that technology must ensure in a database include confidentiality, availability, and integrity. With regards to Oracle, patching is one of the principal best practices. Often database patching and upgrading is not an easy task. Installation of Oracle patches is difficult, especially since systems often require rebooting after patching. Not having a proper patching strategy is due to several reasons such as it is time-consuming, complex, introduces risks, not always really necessary, and also leads to human errors. The database is a center around which many business functions revolve so downtime is a risk and is a drawback to business continuity as well. Oracle recommends applying the patches quarterly, as soon as they are released (Ludovico, 2018). Yet due to the above reasons, companies tend to divert from these patch applications. Therefore, it is necessary to consider implementing a clear patching strategy for the DB.

Legacy Installations is a security failure. Legacy systems lack the advancements in password management, encryption, separation of DBA roles, auditing, provisioning user accounts, etc. Some have interdependencies between applications and databases and rely on external services for security; all of which cause loopholes inside which advanced anti-cybersecurity mechanisms can lurk. To protect the systems against ever-developing attacking mechanisms the database should always be updated timely.

In ORACLE it is needed to get rid of default passwords. One of the most important components of the oracle database is the Listener. The Oracle listener is a service that runs on the database host and receives requests from Oracle clients. One way to protect the Listener is to set the TNS Listener password. In Oracle Database 12c Release 2 (12.2), the listener password feature is no longer supported. This does not cause a loss of security because authentication is enforced through local operating system authentication (Achacoso et al., n.d.). Another security failure occurs at provisioning accounts and privileges. To avoid new account creation by unauthorized users in the database, the company must define naming standards for the accounts that will be used. That way, the DBA will know if a user has been created without permission. It is essential to mention that building a proper access provisioning strategy is very important to run a database healthy and efficiently.

Password is a key to any system. Managing passwords hence become a major part of the security mechanisms of the company. When a database is created, default accounts are created with default passwords. It is needed to change these to unique passwords to secure the database. Also, DBAs need to set the user profile limits such as failed login attempts, password lock times, password reuse, etc. to limit the access to database resources properly and protect them from unnecessarily getting exposed. In addition to these, disabling and enabling the default password security settings, locking inactive database user accounts, automatically locking user accounts after failed logins, controlling the user ability to reuse previous passwords, controlling password aging and expiration, are other security measures related to password management the database engineers can adhere to.

Network security must be addressed for the company’s database as the data being circulated among different roles such as managers and the administrator. Therefore, considering the use of Oracle Advanced Security add-on is suggested here, as it is an option to Oracle Enterprise Edition Servers, which enables support for SSL connections. It also supports encryption and data integrity checks. Oracle Advanced Security provides the Advanced Encryption Standard (AES), DES, 3DES, and RC4 symmetric cryptosystems for protecting the confidentiality of network traffic.

Best practices in SQL programming also play a major role in database health. The DBA is responsible for performing SQL code reviews and walkthroughs to improve DBMS efficiency and performance speed. Therefore, having code review meetings regularly is a best practice.

Planning is a strategic move in any field. Planning for proper backup of data for each database file in the company context is crucial as the database holds important customer personal data and their complaints. Since the database is centralized, backup mechanisms and data migration strategies are critical aspects of the business continuity of the company as the database is at the risk of a single point of failure breakdown.

Here we suggest security and backup strategy for the company's centralized database.

Firstly, the database server should be separated from the webserver. This will ensure the security of both the webserver and the database server. Even if an intruder breaks into the web server, they are unable to access the database server, so that the data is secured. It is best to keep unrelated programs separate from servers. If the programs need to communicate with the servers, it could be enabled through secure network connections. Secondly, the user accounts and access privileges to the database server should be managed with a proper strategy. The DBAs together with technical managers should sit together and decide on a proper long-term user provisioning strategy. Placing a database firewall and a web application firewall is the next concern. Firewalls allow traffic only from specific applications that need to access data. The web application firewall in place prevents attacks to the webserver and also deleting and collecting data from the database through web apps.

The next major step is a plan for backups. When planning for the backup and recovery it is needed to anticipate the risks and errors possible beforehand, to identify the best backup strategies. Oracle provides several solutions for backup and recovery. The suitable choice for the company scenario would be to use Oracle Recovery Manager (RMAN) - the primary utility for physical backup and recovery of Oracle databases (Weill et al., n.d.). RMAN has backup encryption capabilities which we suggest here to use for the company helpdesk data backup. Incremental backups store blocks of changes since the previous backup which is different from differential backups where it stores the whole changed files since the previous backup. Incremental backups are compact thus provide faster recovery. As the helpdesk daily collects a huge amount this can reduce the backup load and help faster recovery in failures. A technique like fallback tables is not workable here as it duplicates entire tables which doubles storage space and I/O for tables.

Backups and mirroring are closely related but they differ in functionality. The suggestion for this strategy is to use both backing up and mirroring as recovery mechanisms for the company database. Mirroring keeps the systems up and running in the event of hard disk failure, backup provides full data protection and recovery capabilities if the whole drive becomes inaccessible. Simply, backups will let you have your old data back. Mirroring just gives you whatever you have now. In case of a malware attack to a database, all the mirrors will include the malware. In such a case, backups help to restore data to servers. Oracle Automatic Storage Management (Oracle ASM) provides software-based mirroring capabilities. Together with RMAN ASM can create the mirrors and backups for the database. RMAN is a feature of the Oracle DB server. Oracle ASM should be installed separately. One major drawback that still exists in the system is the helpdesk database is the database is centralized while the DBA is based at head office. In any situation of a database failure, DBA will take time to do onsite investigations. To overcome the single point of failure risk and remote access challenge, it is best to keep the backups in cloud storage. So, the suggestion is to use Oracle Database Backup Cloud Service which is a secure, scalable, on-demand storage solution for backing up Oracle databases to Oracle Cloud. Using RMAN commands it can execute actions on the Oracle cloud.

The right fallback strategy depends on whether the database can allow some amounts of data to be missing in the backup as the DB team can redo any newer changes. If so, it can be configured in the Oracle to fall back into a particular timestamp. If not, the fallback strategy needs to use the Oracle export (unload) and import (load) to a different Oracle database. The right fallback strategy in this context would be to use an export and import plan. The next part of the strategy is to encrypt all files and backups. There is always a risk that external or internal party getting into the database system. The best way to secure data is to encrypt them all so that any intruder that breaks in cannot read data without an encryption key. This will complete the backup and recovery planning strategy for the company database.

Among all the failures subjected to the database, media failure and user errors are the most critical ones that need an intervention of a DBA. A media failure is a physical problem in the disk that happens when reading or writing data. A common example is a head crash. With the proposed strategy, the control files and redo log files are multiplexed at the Oracle level so if one media fails there are other files multiplexed in different disks so a DBA can recover those files. If the problem is temporary, a crash recovery can be performed using redo log files. If not have to restore a full backup of the entire database. As mirroring maintains two copies of DB in two servers the recovery is possible. In 2009 Microsoft Sidekick servers failed due to problems with device restarts by customers (multipage, 2019). The company had made a great mistake by not keeping proper backups of data so that all customer data were truly lost. In such a scenario mirroring would have been a good solution. Dropping or deleting parts or entire blocks of data are considered user errors. Recovering from such a failure is possible by relocating the lost object from a backup. Keeping a copy of the database using mirroring techniques again can help recover in such a failure as well.

**References**

Achacoso, R., Dadhich, A., Datta, S., Ding, S., Khan, F., Mathur, B., McKinley, S., Pamu, S., Patel, K., Purayathu, M., Ramasubramanian, S., Reguna, S., Woo, N., Jayathirtha, B., n.d. Configuring and Administering Oracle Net Listener [WWW Document]. Oracle Help Center. URL https://docs.oracle.com/en/database/oracle/oracle-database/19/netag/configuring-and-administering-oracle-net-listener.html#GUID-C3C40DBA-4282-41E1-9562-4B8B10947C4E (accessed 4.10.21).

Litchfield, D., 2007. Oracle forensics part 5: Finding evidence of data theft in the absence of auditing.

Ludovico, 2018. Oracle Home Management – part 1: “Patch soon, patch often” vs. reality | DBA survival BLOG. URL http://www.ludovicocaldara.net/dba/oh-mgmt-1/ (accessed 4.9.21).

multipage, 2019. 4 real-life examples of data loss [WWW Document]. Stronghold Data. URL https://www.strongholddata.com/4-real-life-examples-of-data-loss/ (accessed 4.20.21).

Tripathi, S., Meshram, B.B., 2012. Digital Evidence for Database Tamper Detection 2012. https://doi.org/10.4236/jis.2012.32014

Weill, K., Ashdown, L., Bednar, T., Beldalker, A., Chien, T., Dilman, M., Fogel, S., Guzman, R., Haisley, S., Hu, W., Hwang, A., Joshi, A., Krishnaswamy, V., Lee, J.W., Moore, V., Olagappan, M., Panteleenko, V., Ranganathan, S., Sanchez, F., Smith, K.D., Srihari, V., Susairaj, M., Stewart, M., Wertheimer, S., Yang, W., Zijlstra, R., Potineni, P., n.d. Introduction to Backup and Recovery [WWW Document]. Oracle Help Center. URL https://docs.oracle.com/en/database/oracle/oracle-database/21/bradv/introduction-backup-recovery.html#GUID-014F80B5-9A80-4CDB-B282-3FD0C3610FC9 (accessed 4.20.21).