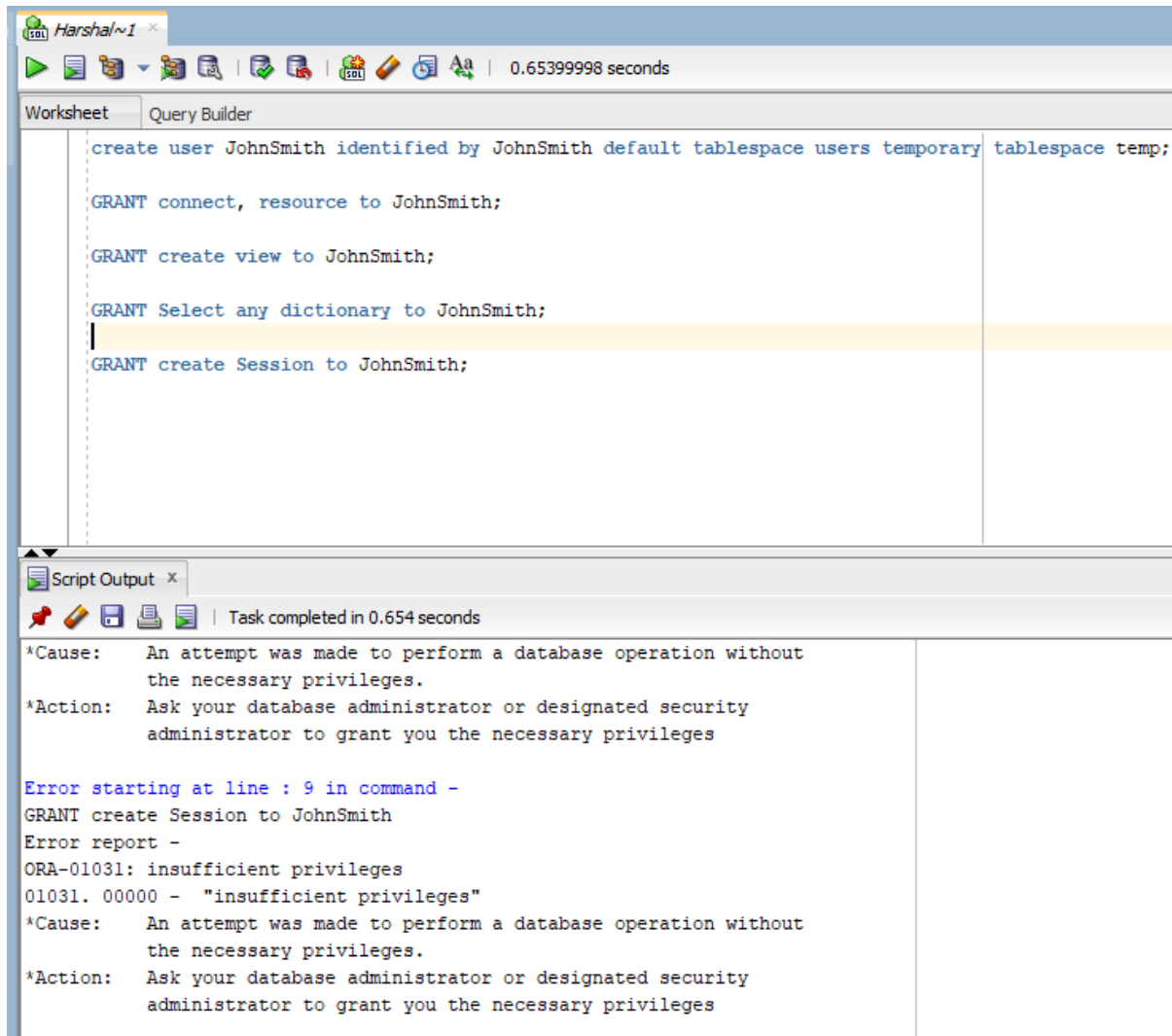


Lab 4 Submittal

Step 1: Creating a User Accounts



```
create user JohnSmith identified by JohnSmith default tablespace users temporary tablespace temp;

GRANT connect, resource to JohnSmith;

GRANT create view to JohnSmith;

GRANT Select any dictionary to JohnSmith;

GRANT create Session to JohnSmith;
```

Script Output x

Task completed in 0.654 seconds

*Cause: An attempt was made to perform a database operation without the necessary privileges.

*Action: Ask your database administrator or designated security administrator to grant you the necessary privileges

Error starting at line : 9 in command -

GRANT create Session to JohnSmith

Error report -

ORA-01031: insufficient privileges

01031. 00000 - "insufficient privileges"

*Cause: An attempt was made to perform a database operation without the necessary privileges.

*Action: Ask your database administrator or designated security administrator to grant you the necessary privileges

The aforementioned command is not run because of inadequate rights, however if it is The instructions provided will establish a user, give them permissions, and configure their temporary and default tablespaces. These instructions typically result in no output or confirmation messages when executed. There won't be any output if the instructions are followed correctly, but a new user named "John Smith" with the necessary rights will be established.

Step 2: Examining User Accounts

2A)

Connection Name | Connection Details
Demo29 | ora_demo29@/...
Harshal | ora_hswant1@/...

Name: Harshal

Database Type: Oracle

User Info | Proxy User

Authentication Type: Default

Username: ora_hswant1 | Role: default

Password: | Save Password: ☐

Connection Type: Basic

Details | Advanced

Hostname: www.papademas.net

Port: 1521

☒ SID: ord

☐ Service name:

Status :

Help Save Clear Test Connect Cancel

Present given role

Connection Name | Connection Details
Demo29 | ora_demo29@/...
Harshal | ora_hswant1@/...

Name: Harshal

Database Type: Oracle

User Info | Proxy User

Authentication Type: Default

Username: ora_hswant1 | Role: SYSOPER

Password: | Save Password: ☐

Connection Type: Basic

Details | Advanced

Hostname: www.papademas.net

Port: 1521

☒ SID: ord

☐ Service name:

Status : Failure -Test failed: ORA-01017: invalid username/password; logon denied

Help Save Clear Test Connect Cancel

We Get an error after changing roles.

Since the error ORA - 01017 denotes an incorrect username or password. Either the grant statement does not contain the ID and password, or we are not allowed to connect to the designated role.

The major reason is that we don't have the necessary privileges to change the present role.

So, we should contact the database administrator to grant desired role the appropriate privileges to connect to the database.

2B)

Users can connect to Oracle databases using a variety of connection types provided by Oracle SQL Developer:

Basic: The most basic connection type, which needs a hostname, password, service name or SID, port, and username.

TNS Connection: This connection type connects to the database using Oracle's Transparent Network Substrate (TNS). It needs a TNS entry, a network configuration that indicates the location of the database. In the connection settings, you provide the name of the TNS entry.

LDAP Connection: You can connect to the database via LDAP (Lightweight Directory Access Protocol) connections by using LDAP directories for service location and authentication. You supply the necessary credentials and the information about the LDAP server.

Cloud Connection: With Oracle Cloud databases, this connection type is appropriate. Usually, it entails entering the username, password, and Oracle Cloud Database Service Name. Connecting to databases hosted on Oracle Cloud Infrastructure is appropriate with this.

SSH Connection: You can safely connect to a distant server and access the database from there via an SSH (Secure Shell) connection. You give your database credentials and SSH server information.

HTTP: When connecting via an HTTP proxy server, this connection type is utilized. When working with databases in settings where a proxy server is being utilized for network access, it comes in handy.

Proxy: Using a proxy connection, you may access the database as a different user.

Advanced connection types: they are intended for particular uses, such as MySQL, Exadata, and other non-Oracle databases.

SID vs. Service Name: To identify the database, pick between the service name and SID; the service name is the recommended choice.

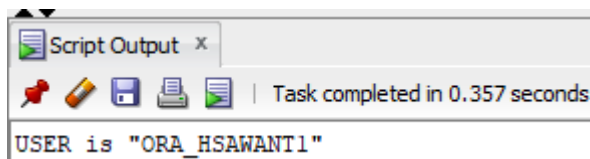
Step 3: Checking Oracle User Management

3-1. Check all users inside the database.

```
Error starting at line : 1 in command -
Select username, account_status, default_tablespace from dba_users
Error at Command Line : 1 Column : 57
Error report -
SQL Error: ORA-00942: table or view does not exist
00942. 00000 - "table or view does not exist"
*Cause:
*Action:
```

The dba_user table does not exist, according to the output above.
The "username," "account_status," and "default_tablespace" columns are selected from the "dba_users" table in an Oracle database by the query you supplied. The information about user accounts, including their default tablespace and account status, is retrieved by this query.

3-2. Check current user.



It results the current user

3-3. Lock/Unlock User

```
Error starting at line : 1 in command -
Alter user username account lock
Error report -
ORA-01031: insufficient privileges
01031. 00000 - "insufficient privileges"
*Cause:    An attempt was made to perform a database operation without
            the necessary privileges.
*Action:   Ask your database administrator or designated security
            administrator to grant you the necessary privileges
```

Lock user

```
Error starting at line : 1 in command -
Alter user username account unlock
Error report -
ORA-01031: insufficient privileges
01031. 00000 - "insufficient privileges"
*Cause:      An attempt was made to perform a database operation without
              the necessary privileges.
*Action:     Ask your database administrator or designated security
              administrator to grant you the necessary privileges
```

Unlock User

Both queries won't work as we don't have the privileges.

3-4. Create New User

```
Error starting at line : 1 in command -
CREATE USER usr1 identified by usr1
Error report -
ORA-01031: insufficient privileges
01031. 00000 - "insufficient privileges"
*Cause:      An attempt was made to perform a database operation without
              the necessary privileges.
*Action:     Ask your database administrator or designated security
              administrator to grant you the necessary privileges
```

Here we are getting the same error of not having the necessary privileges.

3-5. Create new user by assigning a default tablespace.

```
Error starting at line : 1 in command -
CREATE USER usr2 identified by usr2 default tablespace users
Error report -
ORA-01031: insufficient privileges
01031. 00000 - "insufficient privileges"
*Cause:      An attempt was made to perform a database operation without
              the necessary privileges.
*Action:     Ask your database administrator or designated security
              administrator to grant you the necessary privileges
```

Here we are getting the same error of not having the necessary privileges.

3 - 6. Change user password.

```
Error starting at line : 1 in command -
ALTER USER Demo29 identified by abcd
Error report -
ORA-01031: insufficient privileges
01031. 00000 - "insufficient privileges"
*Cause:      An attempt was made to perform a database operation without
              the necessary privileges.
*Action:     Ask your database administrator or designated security
              administrator to grant you the necessary privileges
```

Here I have tried to change the password of another user but due to insufficient privileges, I'm unable to perform that action.

3 -7. Create more allocable space.

```
Error starting at line : 1 in command -  
ALTER USER Demo29 quota 100M on users  
Error report -  
ORA-01031: insufficient privileges  
01031. 00000 - "insufficient privileges"  
*Cause:      An attempt was made to perform a database operation without  
              the necessary privileges.  
*Action:     Ask your database administrator or designated security  
              administrator to grant you the necessary privileges
```

Due to a lack of privilege, the aforementioned query is not executed. To provide precise quotas for a user on a certain tablespace, use the query shown above. For this query, a 100 MB limit has been established. This helps control resource consumption and guarantee that users don't utilize more tablespace than they have allocated.

Step 4: Tablespace Quota

```
Error starting at line : 1 in command -  
ALTER USER usr1 quota 20M on users  
Error report -  
ORA-01031: insufficient privileges  
01031. 00000 - "insufficient privileges"  
*Cause:      An attempt was made to perform a database operation without  
              the necessary privileges.  
*Action:     Ask your database administrator or designated security  
              administrator to grant you the necessary privileges
```

The aforementioned command will generate a storage quota on the USERS tablespace for user 1.

The user "usr1" for whom we have established a quota is mentioned in this area of the phrase. 20 QUOTA M This represents the 20 M user limit. Maximum allowed storage space in 20 M ABOUT USERS The statement's target tablespace, to which the quota is applied, is shown here. Quota is applied to the USERS tablespace in this case.

Step 5: Investigate Roles and Profile Management.

5a) creating roles and profile accounts

```
Error starting at line : 1 in command -  
CREATE ROLE SALES_MANAGER  
Error report -  
ORA-01031: insufficient privileges  
01031. 00000 - "insufficient privileges"  
*Cause:      An attempt was made to perform a database operation without  
              the necessary privileges.  
*Action:     Ask your database administrator or designated security  
              administrator to grant you the necessary privileges
```

We got the same error of insufficient privileges, unless it would have created a role of SALES_MANAGER.

5b) Creating and administering rules for password protection

To create and administer rules for password protection in Oracle SQL Developer, you can follow these steps based on the information provided:

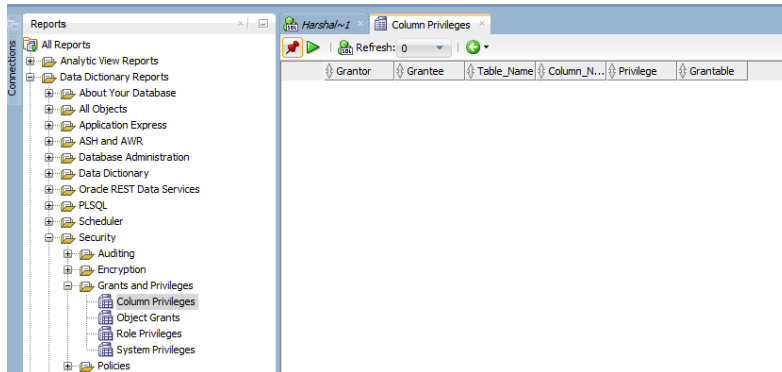
1. Create a Role:
 - Use the 'CREATE ROLE' statement to create a role with the desired name.
 - Optionally, use 'IDENTIFIED BY password' to create a local role requiring a password, or use 'NOT IDENTIFIED' for a role authorized by the database.
2. Grant Privileges to the Role:
 - Use the 'GRANT' statement to assign system or object privileges to the created role.
 - For instance: 'GRANT {system_privileges | object_privileges} TO role_name;'
 - You can also grant a role to another role using: 'GRANT role_name TO another_role_name;'
3. Example without Password Protection:
 - Create a role "mdm."
 - Grant object privileges on tables like customers, contacts, etc., to the "mdm" role.
4. Create a User:
 - Create a user, e.g., "alice," and grant the 'CREATE SESSION' privilege to allow login.
5. User Login
 - Log in as "alice" and attempt to access data, which will initially result in an error.
6. Assign the Role to the User
 - Grant the "mdm" role to "alice."
7. Enable the Role
 - In the user's session, use 'SET ROLE' to enable the "mdm" role.
8. Check Current Roles
 - Use 'SELECT * FROM session_roles;' to verify the active roles.

These steps allow you to create roles, assign privileges, and manage user access with Oracle SQL Developer, demonstrating how to control access and authorization effectively.

But, in order to apply all these permissions, you need to have the role of a DBA, or should have all the permissions granted.

Step 6. The Security Folder in SQL Developer.

6a) Go the Grants and Privileges folder and drill down to the Column Privileges and take a snapshot of what is given and paste it in your submittal document.



Users may control and view the rights given to individual columns in a database table by visiting the column privileges section. For any of the aforementioned columns in the table, there are neither stated nor granted explicit column level permissions.

6b) Go the Public Grants folder and drill down to the Counts by Grantor and take a snapshot of what is given and paste it in your submittal document.

The screenshot shows the SQL Developer interface. On the left, the 'Connections' pane is open, showing a tree view of the database structure. The 'Security' folder is expanded, and 'Public Grants' is selected. The main pane displays the 'Counts by Grantor' report, which shows a list of grantors and the count of grants for each.

GRANTOR	COUNT_OF_GRANTS
1 ADMIN_RAZMEERA	1
2 APEX_040200	250
3 BI	1
4 CTXSYS	122
5 DVF	17
6 DVSY	10
7 GSHADMIN_INTERNAL	13
8 GSHCATUSER	1
9 IX	1
10 LBACSYS	43
11 MDSYS	1184
12 MM	1
13 MM2	1
14 OLAPSYS	19
15 ORA_AANAND24	1
16 ORA_AANAND25	1
17 ORA_AAYYALUKUMARAN	1
18 ORA_ABARRE	1
19 ORA_ACOELHO	1
20 ORA_AELLANDULA	1
21 ORA_AGAJBAHAR	1
22 ORA_AGANJ11	1
23 ORA_AKAMENOVIC	1
24 ORA_AMALI3	1
25 ORA_APATIL62	1
26 ORA_APIMPLE1	1
27 ORA_AYAMEN	1
28 ORA_AYASANGI	1
29 ORA_BAATHAVA	1

The number of privileges or grants made available to the general public in the Oracle database is displayed in the above result. It shows how many grants are made to the public, what kinds of rights are awarded, what is affected, security and audits, and compliance data access control.

6c) What is meant by encryption and how can it be used for a table? Give an example.

Data is transformed into a code through the process of encryption to prevent unwanted access or reading. Encryption may be used in Oracle SQL Developer to safeguard data kept in database tables and keep private information out of the hands of unauthorized individuals. Transparent Data Encryption (TDE) may be used in Oracle SQL Developer to encrypt a table. Data is encrypted on disk when using Oracle Database's TDE functionality, which encrypts data while it's at rest.

Here's a condensed illustration of how to use Oracle SQL Developer to activate TDE for a table:

- Open SQL Developer and connect to your Oracle database.
- Verify that TDE (Transparent Data Encryption) is turned on in the database.
- Make a table that contains the sensitive information you wish to encrypt.
- CREATE TABLE employee (emp_id NUMBER, emp_name VARCHAR2(50)); as an example
- For the tablespace in which the table is stored, enable TDE. This may be accomplished by managing encryption keys using the Oracle Wallet.
- Choose the tablespace to be encrypted in order to encrypt the table. ALTER TABLESPACE employee_tablespace ENCRYPT, for instance

Step 7. SQL Injection Threats.

7a) An IPS (intrusion prevention signature) system has been suggested to the IT network admin by the DBA to be purchased, installed and deployed. The DBA fears that, with the company's newly implemented website that will allow back-end entrance into sensitive data, sql-injections will be imminent. The IT network admin quickly performs research into IPS systems. What information will be revealed?

In order to defend against any SQL injection attacks, the IT network administrator would look at IPS (Intrusion Prevention System) settings and find the following important information:

Objective of an IPS: The aim of an intrusion prevention system (IPS) is to stop hostile activity by monitoring network traffic, stopping attempted intrusions, and stopping vulnerabilities from being exploited.

Threats from SQL Injection: The DBA is right to be concerned about SQL injection assaults. Attackers frequently employ SQL injection as a way to take advantage of holes in online applications and possibly access databases or private information without authorization.

Characteristics of an IPS:

Signature-based Detection: In order to recognize established attack patterns, such as SQL injection assaults, intrusion prevention systems (IPS) often employ predetermined signatures or patterns.

Psychological Analysis: To spot unusual traffic patterns that may point to a SQL injection attempt, certain contemporary IPS systems employ behavioral analysis.

Surveillance : Intrusion Prevention Systems (IPS) keep an eye on network traffic to spot questionable activity and react quickly with notifications.

Limiting Capabilities: By deleting or altering harmful packets, intrusion prevention systems (IPS) can prevent or lessen attacks.

Options for Deployment:

-On already-existing network infrastructure, IPS may be installed as software or as a hardware device.

-Certain systems provide scalability and flexibility through cloud-based or hybrid deployments.

Modification: IPS systems frequently let administrators modify rules and policies to better suit the unique requirements of their network.

False Positives: Be mindful of false positives since improperly setup, overly vigilant intrusion prevention systems might block valid communications.

Alternatives for Vendors: Different vendors provide IPS systems with varying features and costs. The administrator would have to look into and select a vendor who best meets the needs of the business and its budget.

Cost factors: IPS systems could include one-time or recurring maintenance fees, upfront charges for hardware or software licensing, and sometimes even operating costs.

Integration: The selected intrusion prevention system (IPS) need to blend in perfectly with the current network and security setup, as well as the most recent website launch and its backend elements.

Efficiency Impact: Since IPS systems may cause some delay, the administrator should take into account any possible effects on network performance.

Regulatory Adherence: The IPS may need to handle certain compliance standards (like GDPR or HIPAA) based on the industry and data being secured.

Instruction and Documentation: Make sure that the IT staff is adequately trained to setup and administer the intrusion prevention system (IPS) and that documentation about best practices and troubleshooting is available.

Testing and Validation: To make sure the IPS successfully identifies and thwarts SQL injection attempts without interfering with regular operations, it should be extensively tested in a monitored setting prior to deployment.

After compiling this data, the IT network administrator may decide with confidence which intrusion prevention solution to install and use in order to protect the company's network against SQL injection attacks. (*Nist Special Publication 800-53 Revision 4 - Csf Tools*, n.d.)

7b) The company's worst fears were realized as an SQL-injection occurred and sensitive client information was stolen. Are there steps, and therefore what are the steps, should the company undertake to alleviate the theft?

It is critical to respond quickly to minimize damage and stop future breaches when a firm is the victim of a SQL injection attack and sensitive client information is taken. The following actions are what the business should do to lessen the theft:

Isolate the Compromised System: To prevent more breaches, isolate the impacted system right away.

Assess and stop the attack: Put an end to the attack by fixing the SQL injection vulnerability to stop further exploitation.

Contact impacted Clients: Advise impacted clients on preventive measures and promptly notify them.

Legal Compliance: Adhere to the rules and legislation pertaining to data breach notification.

Analysis: Examine the breach to comprehend the attack and collect evidence using forensic analysis.

Improve Security: To stop future assaults, put in place more robust security measures.

Password Changes: Inform impacted clients to update their passwords.

Constant Monitoring: Establish a real-time watch for questionable activity.

Backup and Recovery of data: Verify that there exist backup data sets that can be restored.

Post-attack Review: Enhance security and draw lessons from the intrusion.

Staff awareness session: Provide security awareness training to staff members. (Bertino & Sandhu, n.d.)

7c) It was found that access to the company's website had several vulnerabilities, i.e., the contact information web page, and the email address system. Find examples of sql-injection scripts that could have been used to breach security and how the IT network admin AND the DBA can overcome those present vulnerabilities.

SQL Injection: To get unauthorized database access, malicious SQL code is injected in a SQL injection cyberattack.

Example of a script can be:

```
SELECT * FROM users WHERE username = '$username' AND password = '$password';
```

If the validation is poor, then the hacker/attacker can input **admin--** in username through SQL injection process and then he can modify the above query to –

```
SELECT * FROM users WHERE username = 'admin'
```

So, he would be able to login without knowing the password.

Ways to mitigate SQL Injection:

1. Make use of parameterized queries and prepared statements.
2. Put Web Application Firewall (WAF) into action.
3. Verify and clean user input.
4. For database users, adhere to the least privilege concept.
5. Steer clear of lengthy error messages.
6. Audit and run tests for security flaws on a regular basis.

Therefore, to avoid this DBAs and Network admins should obey the following instructions:

For IT Network admins:

1. Keep an eye on network traffic, IT network administrator.
2. Set up firewalls to prevent unauthorized requests.
3. Update the hardware and software on your network.
4. Implement access restrictions.

Database administrators (DBAs) should:

1. Work with developers to ensure that user input is validated.
2. Implement access restrictions and safeguard the database server.
3. Make frequent data backups.
4. Turn on logging and auditing.
5. To identify intrusions, use database security technologies. (Sadeghian et al., 2013)

Step 8. Performing Research

8a) What are some ways that are performed to securing network connections from a client to the database server?

Data safety requires secure network connections between clients and database servers. Key actions to do this are as follows:

1. **Encryption:** To guarantee the secrecy of data while it is being sent, use SSL/TLS.
2. **Firewalls:** Install firewalls in order to limit and regulate network access to the database server. Let only reliable IP addresses.
3. **Network Segmentation:** To reduce exposure, isolate the database server on a different network.
4. **Authentication and Authorization:** To grant access to only authorized users, employ role-based access control in conjunction with strong authentication.
5. **Password Policies:** Don't use default credentials; instead, enforce strong passwords that are cycled often.
6. **Using two-factor authentication (2FA):** Using 2FA to authenticate users can improve security.
7. **IP Whitelisting:** To stop unwanted access, whitelist particular IP addresses or IP ranges.
8. **Data at Rest Encryption:** For further security, encrypt data kept in the database.
9. **Patch management:** Apply security updates to software.
10. **Intrusion Detection and Prevention:** Keep an eye out for questionable activity and stop illegal access.
11. **Audits and logs:** Enable thorough audits and logging in order to monitor actions and access.
12. **Least Privilege:** Restrict user rights to those absolutely necessary to do their responsibilities.
13. **Security Testing:** Perform testing and evaluations on a regular basis.
14. **Backups:** Make regular database backups and store them in a safe location.
15. **Network monitoring:** Keep an eye on database activity and network traffic at all times.
16. **Security policies & training:** Enforcement of Security Policies and Employee Training: Make sure that security policies are followed.
17. **Incident Response:** Have a strategy in place for reacting quickly to security incidents. By combining these strategies, network security is strengthened and the likelihood of breaches and illegal access is decreased. (*Nist Special Publication 800-53 Revision 4 - Csf Tools*, n.d.)

8b) What are some security vulnerabilities and how can they be detected?

Typical security flaws and how to find them include the following:

SQL Injection (SQLi): To access databases without authorization, attackers insert malicious SQL code into input areas. Web application firewalls, parameterized queries, and input validation are all part of detection.

Cross-Site Scripting (XSS): Malicious scripts are inserted into websites that other users see, a technique known as cross-site scripting (XSS). Web vulnerability scanners, content security rules, and input sanitization are examples of detection techniques.

Cross-Site Request Forgery (CSRF): This attack technique fools users into executing commands against their will. Strict referer headers, anti-CSRF tokens, and CSRF vulnerability testing are all used in the detection process.

Inadequate Authentication and Passwords: Unauthorized access might result from insufficient authentication and weak passwords. Multi-factor authentication, frequent password audits, and the enforcement of strong password regulations are all examples of detection.

Security Misconfigurations: Vulnerabilities may be revealed by improperly configured servers or apps. Regular security audits, automated scanning technologies, and adherence to safe setup guidelines are all part of detection.

Insecure Deserialization: Applications are injected with malicious data to execute code. Validating input, keeping an eye on deserialization operations, and utilizing serialization libraries with integrated security measures are all part of detection.

Vulnerabilities related to File Inclusion: An attacker may include files from a distant server. Input validation, appropriate access controls, and preventing user input in file paths are all included in detection.

Information Disclosure: Directory listings or error messages may include sensitive information. Comprehensive error handling, security headers, and routine testing for such leaks are all part of the detection process.

Buffer Overflows: Programs are exploited by attackers using buffer overflows. Code reviews, code analysis tools, and runtime security procedures are examples of detection. (Alvarez et al., 2016)

8c) Safeguarding the system from intruders - How do firewalls and or routers help in preventing intruders/malware data packets from entering the company's database?

By regulating and keeping an eye on network traffic, firewalls and routers are essential components in protecting computer networks against hackers and malicious data packets. They safeguard a company's database and aid in preventing illegal access in the following ways:

Packet filtering: Unauthorized packets are blocked by firewalls and routers using rules to filter incoming and outgoing traffic.

Access Control Lists (ACLs): ACLs limit authorized users' or devices' access to particular resources, such as databases.

Stateful Packet Inspection (SPI): They keep an eye on ongoing connections in order to distinguish between malicious and genuine traffic.

Application Layer Filtering: To detect malware or questionable material, firewalls scan data packets.

IDS/IPS: To identify and address threats, certain devices are equipped with intrusion detection and prevention systems.

Network Address Translation (NAT): As an additional security measure, routers mask internal IP addresses.

VPNs: Routers protect data by encrypting it and enabling secure remote access via VPNs.

Logging and Monitoring: They keep track of network activities in order to spot and fix security vulnerabilities.

Frequent Updates: In order to fix vulnerabilities and enhance security, it's imperative to keep these devices updated.

Security Policies: For network protection, it is essential to enforce explicit security policies and train staff members.

In conclusion, routers and firewalls are essential elements of network security. They serve as walls that separate your company's database from hackers and malicious data packets by screening traffic, enforcing security regulations, and preventing unauthorized access. It is imperative to appropriately install, monitor, and maintain these devices in order to provide efficient protection. (Frahim et al., 2014)

**8d) Will using an Intrusion Detection System (IDS) help in maintaining security?
Support your answer.**

Indeed, employing an intrusion detection system (IDS) may be a useful part of an all-encompassing security plan, but it needs to be utilized in concert with other security measures as a stand-alone solution. Here's how IDS may support security maintenance:

Early Threat Detection: IDS has the ability to detect harmful or questionable activity in real time or very close to it. It warns security staff about possible dangers before they have a chance to do a lot of harm.

Diminished Response Time: Security teams can react quickly to security events when IDS is in place. This can lessen the effects of an incursion or stop it altogether.

Pattern Recognition: To recognize known attack signatures or departures from typical network activity, intrusion detection systems (IDS) employ pattern recognition and anomaly detection. This aids in the identification of different kinds of assaults, including brute force and malware outbreaks.

Minimized False Positives: To ensure that warnings are more accurate, contemporary IDS systems are built to minimize false positives.

Real-time notifications: When it detects suspicious behaviour, IDS may send out real-time notifications, enabling security staff to react swiftly to possible security problems. This pre-emptive action can lessen or stop the harm.

Log and Data Analysis: Intrusion Detection Systems (IDS) gather and examine network and system logs, which can be essential for spotting attack trends and comprehending the strategies employed by possible hackers. Enhancing security measures can be greatly aided by the information provided.

To support my answer above here's an example:

Target experienced a significant data breach in 2013, during which hackers took millions of credit card details from customers. Target had an intrusion detection system in place, but its poor handling of the system's warnings resulted in a serious breach. This emphasizes the necessity of having an IDS in addition to efficient procedures and staff that can react quickly. When used properly, intrusion detection systems (IDS) may improve security by quickly warning businesses about possible threats and weaknesses, allowing for proactive actions. (Frahim et al., 2014)

8e) Is ransomware also a threat to database security? What steps should be taken to prevent ransomware?

Yes, ransomware poses a serious risk to the security of databases. Because databases hold sensitive and important information, hackers find them to be appealing targets. Ransomware attacks on databases can have serious repercussions, such as data loss, financial expenses, and even legal and regulatory problems. You should take the following actions to guard against ransomware attacks on your databases:

Backup Data: Ensure that you regularly create a safe backup of your databases.

Access Controls: Strong access controls and authentication should be put in place.

Software Updates: Apply security fixes to keep databases and related software current.

Network Security: To safeguard your network, use intrusion detection systems and firewalls.

Security software installation: Install antivirus and anti-malware software as security measures.

User Education: Inform employees about the dangers of social engineering and phishing.

Security Policies: Create and implement guidelines and practices for security.

Email Filtering: To prevent harmful material, use email filtering.

Data encryption: Encrypt data both in transit and at rest using data encryption.

Remember, the best defence against ransomware is a multi-layered strategy. (Pagán & Elleithy, 2021)

REFERENCES

1. *Nist special publication 800-53 revision 4—Csf tools*. (n.d.). Retrieved 22 September 2023, from <https://csf.tools/controlset/nist800-53r4/>
2. Bertino, E., & Sandhu, R. (n.d.). *Database security—Concepts, approaches, and challenges*. Retrieved 22 September 2023, from <https://ieeexplore.ieee.org/abstract/document/1416861>
3. Sadeghian, A., Zamani, M., & Manaf, A. Abd. (2013). A taxonomy of sql injection detection and prevention techniques. *2013 International Conference on Informatics and Creative Multimedia*, 53–56. <https://doi.org/10.1109/ICICM.2013.18>
4. Alvarez, D. E., Correa, D. B., & Arango, F. I. (2016). An analysis of XSS, CSRF and SQL injection in colombian software and web site development. *2016 8th Euro American Conference on Telematics and Information Systems (EATIS)*, 1–5. <https://doi.org/10.1109/EATIS.2016.7520140>
5. Pagán, A., & Elleithy, K. (2021). A multi-layered defense approach to safeguard against ransomware. *2021 IEEE 11th Annual Computing and Communication Workshop and Conference (CCWC)*, 0942–0947. <https://doi.org/10.1109/CCWC51732.2021.9375988>