Computers, Materials & Continua

Tech Science Press

DOI: 10.32604/cmc.2024.057673

ARTICLE

Classification of Cybersecurity Threats, Vulnerabilities and Countermeasures in Database Systems

Mohammed Amin Almaiah[1,*], Leen Mohammad Saqr[1], Leen Ahmad Al-Rawwash[1], Layan Ahmed Altellawi[1], Romel Al-Ali[2,*] and Omar Almomani[3]

[1]King Abdullah the II IT School, University of Jordan, Amman, 11942, Jordan

[2]The National Research Center for Giftedness and Creativity, King Faisal University, Al-Ahsa, 31982, Saudi Arabia

[3]Department of Networks and Cybersecurity, Al-Ahliyya Amman University, Amman, 19328, Jordan

*Corresponding Authors: Mohammed Amin Almaiah. Email: m.almaiah@ju.edu.jo; Romel Al-Ali. Email: ralali@kfu.edu.sa

Received: 24 August 2024 Accepted: 25 October 2024 Published: 18 November 2024

ABSTRACT

Database systems have consistently been prime targets for cyber-attacks and threats due to the critical nature of the data they store. Despite the increasing reliance on database management systems, this field continues to face numerous cyber-attacks. Database management systems serve as the foundation of any information system or application. Any cyber-attack can result in significant damage to the database system and loss of sensitive data. Consequently, cyber risk classifications and assessments play a crucial role in risk management and establish an essential framework for identifying and responding to cyber threats. Risk assessment aids in understanding the impact of cyber threats and developing appropriate security controls to mitigate risks. The primary objective of this study is to conduct a comprehensive analysis of cyber risks in database management systems, including classifying threats, vulnerabilities, impacts, and countermeasures. This classification helps to identify suitable security controls to mitigate cyber risks for each type of threat. Additionally, this research aims to explore technical countermeasures to protect database systems from cyber threats. This study employs the content analysis method to collect, analyze, and classify data in terms of types of threats, vulnerabilities, and countermeasures. The results indicate that SQL injection attacks and Denial of Service (DoS) attacks were the most prevalent technical threats in database systems, each accounting for 9% of incidents. Vulnerable audit trails, intrusion attempts, and ransomware attacks were classified as the second level of technical threats in database systems, comprising 7% and 5% of incidents, respectively. Furthermore, the findings reveal that insider threats were the most common non-technical threats in database systems, accounting for 5% of incidents. Moreover, the results indicate that weak authentication, unpatched databases, weak audit trails, and multiple usage of an account were the most common technical vulnerabilities in database systems, each accounting for 9% of vulnerabilities. Additionally, software bugs, insecure coding practices, weak security controls, insecure networks, password misuse, weak encryption practices, and weak data masking were classified as the second level of security vulnerabilities in database systems, each accounting for 4% of vulnerabilities. The findings from this work can assist organizations in understanding the types of cyber threats and developing robust strategies against cyber-attacks.

KEYWORDS

Cyber threats; database systems; cyber risk assessment; vulnerabilities; countermeasures

1 Introduction

Database management systems (DBMS) constitute a critical component of Information Technology (IT) infrastructure in any information system, primarily due to their role in storing sensitive data [1]. These systems are essential for maintaining customer data and transaction records. Consequently, organizations have become increasingly reliant on DBMS due to their substantial benefits. However, this dependence has introduced new challenges related to cybersecurity risks and attacks. Currently, cyber threats represent the most significant challenges facing database management systems, with the number of cyber-attacks on these systems increasing and becoming more sophisticated [2]. Cyber risks in DBMS can have severe consequences, including data loss, reputational damage, and system failure. Therefore, it is crucial to understand the behavior of cyber threats on database systems and identify appropriate countermeasures to mitigate their impacts [3]. In the contemporary digital landscape, database systems serve as the backbone of modern IT society, supporting various applications such as business operations, scientific research, and technological innovation. DBMS offers numerous advantages, including ease of data storage, retrieval, modification, and deletion, as well as various data processing operations. As database management systems continue to evolve and expand, they face significant challenges due to emerging attacks that threaten their security [4].

While database systems offer numerous advantages, including enhanced information quality, consistency, accessibility, and efficiency, they are also more susceptible to cybersecurity attacks. The widespread adoption of database systems in organizations has given rise to new cybersecurity vulnerabilities that can be exploited. In the database field, cybersecurity attacks such as SQL injection, distributed denial-of-service (DDoS), and ransomware have become increasingly prevalent, posing significant risks [5,6]. The continuous development of new attack techniques by cybercriminals presents substantial challenges that require addressing [7,8]. Consequently, database security analysts must continuously assess security threats to detect emerging risks and protect database systems and their data from unauthorized modifications. Organizations must remain vigilant about potential threats to their databases, comprehend their impacts, implement preventive measures, and mitigate their negative consequences. Furthermore, they should identify and address vulnerabilities in their systems and devices promptly upon discovery, striving to maintain data confidentiality, integrity, and availability. The most common threats to database systems include insider threats, SQL injections, phishing, and DDoS attacks [7,8].

Previous studies have highlighted several security issues in database systems, including inadequate encryption, insufficient access controls, and outdated software [9,10]. These vulnerabilities may lead to data corruption, unauthorized access, and service interruptions. Database security threats are defined as exploitable weaknesses that compromise the confidentiality, integrity, and availability of stored data. In recent years, numerous cybersecurity attacks have been witnessed on database systems. For instance, Verizon's 2020 Data Breach Investigations Report revealed that 45% of breaches involved hacking, while 22% were attributed to social attacks such as phishing or pretexting. Additionally, an IBM (International Business Machines Corporation) study reported that the average cost of a data breach in 2020 was $3.86 million. Risk assessment can safeguard companies against such financial losses. Therefore, understanding potential threats is crucial in risk assessment and should be considered when developing a robust security strategy to prevent data breaches. Security risk assessment plays a vital role in identifying potential threats, implementing proactive security measures, and mitigating the likelihood of successful attacks. Cybersecurity risk assessment for database systems is an ongoing process rather than a one-time task. By identifying and classifying risks, implementing appropriate security controls, and evaluating their effectiveness, organizations can significantly reduce potential

threats and risks in database systems. Consequently, this research aims to achieve the following objectives:

(1) To identify and categorize the primary cybersecurity threats in database systems.
(2) To identify and classify the principal cybersecurity vulnerabilities in database systems.
(3) To identify and systematize the key cybersecurity countermeasures in database systems.

2 Related Works

Several studies have explored and classified cybersecurity risks and threats in database systems. Omotunde et al. [3] conducted a comprehensive review to identify the main security controls in database systems. They categorized these controls into five groups: privacy-enhancing techniques, Intrusion Detection Systems (IDS), auditing, encryption, and access control and authentication. Touil et al. [6] analyzed critical cyber-attacks in database systems based on blockchain technology. Their findings revealed that the primary cyber-attacks in database systems include DoS, unauthorized access, internal threats, black hat activities, social engineering, SQL injection, and abuse of excessive privilege. Similarly, Pan et al. [5] investigated the main cyber threats in database systems and identified SQL injection, cross-site scripting, data leakage, and malware as the primary threats that attackers can exploit to compromise database systems. Teimoor [4] conducted a study to identify cyber threats, risks, and countermeasures in databases. The research classified cyber-attacks into two main types: passive attacks and active attacks. Passive attacks encompass static leakage, outflow of information, and dynamic leakage, while active attacks include spoofing, splicing, and replay. The study also identified five key countermeasures to protect databases: access control, inference strategy, user authentication, accountability and auditing, and encryption techniques.

3 Research Design and Framework

This section outlines the research design for proposing a risk assessment framework for database systems. The framework comprises four primary stages: (1) identifying key components, (2) threat identification, (3) vulnerability identification, and (4) countermeasure identification. Each stage is informed by the findings from the literature review. The primary objective of this risk assessment framework is to provide a robust and comprehensive approach for addressing all types of threats, vulnerabilities, and countermeasures in database systems. Fig. 1 illustrates the main stages of the risk assessment framework.

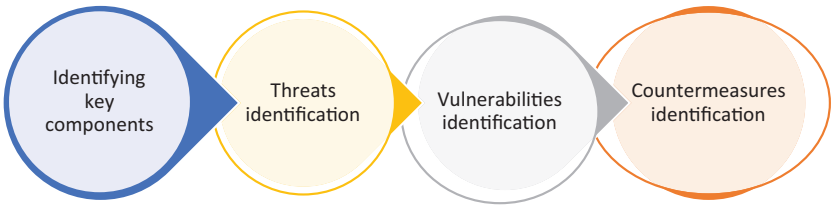

Figure 1: The main stages of risk assessment framework

3.1 Stage One: Identifying Key Components

The initial phase of the risk assessment framework involves compiling data from literature review findings to establish the dataset for this study. This process entails a comprehensive examination of existing studies, models, frameworks, and literature in the field of security database systems. The

collected data encompasses threat types, vulnerability categories, and countermeasure methodologies. The information gathered during this stage will undergo analysis in subsequent phases.

3.2 Stage Two: Threats Identification

Following data collection in the first stage, the subsequent phase involves analyzing the gathered information to identify and categorize existing cybersecurity threats in database systems. This stage encompasses a comprehensive and systematic process that identifies various types of threats with the potential to exploit vulnerabilities in database systems, potentially resulting in compromised systems.

3.3 Stage Three: Vulnerabilities Identification

In the third stage, following data collection, an analysis is conducted to identify existing technical security vulnerabilities that could potentially compromise database systems. This stage of the risk assessment framework incorporates a comprehensive systematic review to determine critical vulnerability types that may be exploited to breach database systems.

3.4 Stage Four: Countermeasures Identification

The final phase of the risk assessment framework involves identifying and categorizing effective countermeasures to address potential cybersecurity threats and vulnerabilities in database systems. The identification of these countermeasures is directly linked to all types of threats and vulnerabilities identified in the previous stages' findings. Consequently, this stage provides solutions to mitigate potential threats that could compromise the integrity of database systems.

4 Cyber Threats, Vulnerabilities, and Countermeasures Framework

Fig. 2 illustrates the primary components of the research framework. The framework comprises three main parts: (1) threat identification, (2) vulnerability identification, and (3) countermeasure identification. The following subsections will provide detailed explanations of each step within the framework.

4.1 Threats Identification

The initial phase of the framework involves identifying and classifying existing cybersecurity threats in database systems. This step encompasses a comprehensive, systematic classification of all potential threats that could exploit vulnerabilities in database systems, potentially leading to compromised systems. The threat classification is categorized into two main groups: (1) technical threats and (2) non-technical threats. Technical threats encompass malware types that exploit security weaknesses in the IT infrastructure of database systems, such as SQL injection attacks, DDoS attacks, TCP (Transmission Control Protocol) spoofing attacks, and malicious traffic attacks. Conversely, non-technical threats include insider threats, bypass/physical attacks, human errors, and illegal user behavior. The classification analysis is based on multiple dimensions, including threat characteristics, behaviors, and their impacts. Each threat type is described with an explanation of its potential impact on database systems. The subsequent subsections provide a detailed threat classification of both technical and non-technical threats.

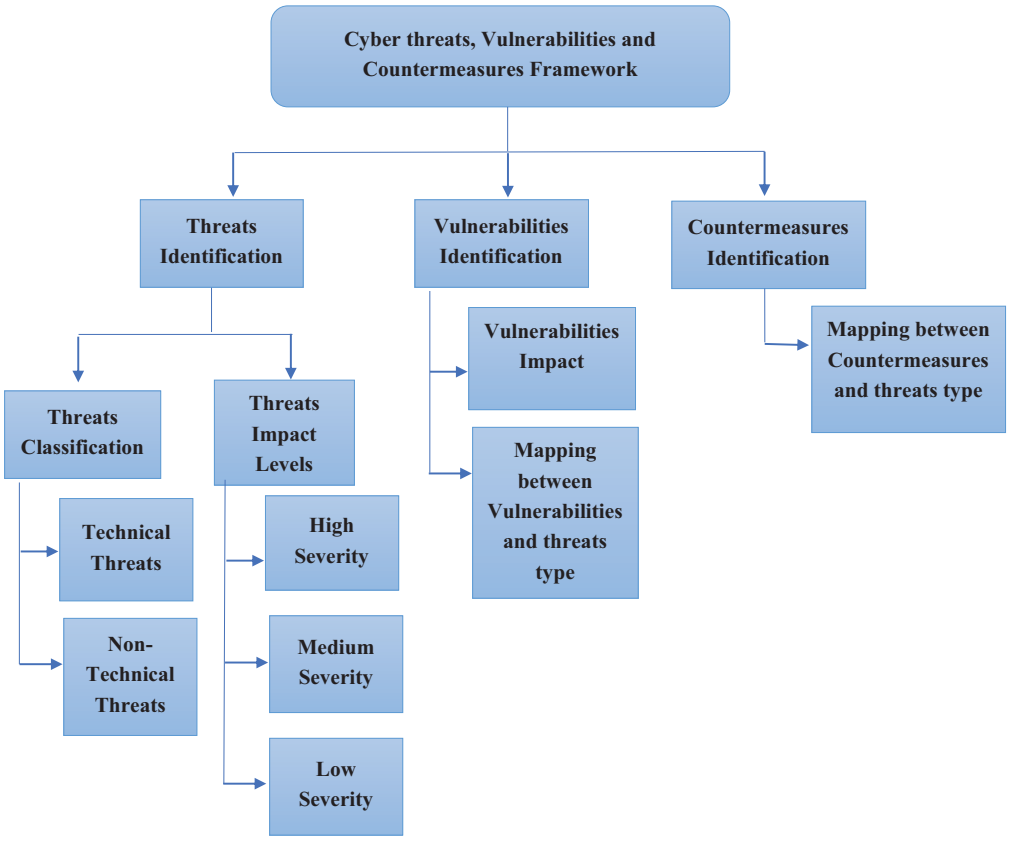

Figure 2: Cyber threats, vulnerabilities and countermeasures framework

4.1.1 Classification of Technical Threats

As discussed in the preceding section, we categorize cyber threats in database systems into technical threats, as illustrated in Table 1. Technical threats encompass various malware types that exploit security vulnerabilities in the IT infrastructure of database systems, including SQL injection attacks, DDoS attacks, TCP spoofing attacks, and malicious traffic attacks. According to Table 1, ransomware is a specific type of malware that restricts access to the victim's data through encryption and demands a ransom payment for restoring access. There are two primary types of ransomware attacks on databases: (1) Encryption ransomware, which utilizes built-in database functions and methods such as transparent data encryption or traditional encryption standards like Advanced Encryption Standard (AES), Rivest–Shamir–Adleman (RSA), and Data Encryption Standard (DES) to encrypt data before writing it to a disk, and (2) Exfiltration ransomware, whose primary objective is data theft. Attackers may employ database dumping tools, evasion techniques such as Domain Name System (DNS) exfiltration to evade detection and circumvent security controls, and utilize SELECT queries.

Table 1: Classification of technical threats in database systems

Technical threats	Description of threats	Impact and example of threats
Phishing [1,8]	Phishing is based on social engineering and is defined as a fraudulent attack because an attacker convinces the victim to do something harmful to the system or to themselves by providing sensitive data to the attacker. Usually, the attacker's intent is to steal system credentials for financial gain or to execute other attacks, such as ransomware. Attackers perform phishing over three mediums: the Internet, messaging services, and voice. However, short messaging services and voice are attacked by smishing or vishing.	Spear phishing and clone phishing
Malware [3,7,8]	Ransomware can encrypt database files, rendering them inaccessible until a ransom is paid. Two types of ransomware are Encryption ransomware and Exfiltration ransomware.	Ransomware
Attacks on graphical passwords [3,10]	A hacking method that exploits weaknesses and vulnerabilities using trial and error or other ways to crack passwords to gain unauthorized access to the database and user accounts and data.	Shoulder surfing Video-recording attack Smudge attack Spyware attack Brute-force attack Computer vision attacks Dictionary attacks Eavesdropping The frequency of occurrence analysis attack Social engineering attacks Image gallery attacks Statistical attacks
Intrusion attempts [1,6,7]	A threat is when intruders try to overcome existing security measures and gain unauthorized access to a system, which leads to breaches, losses, and damage to an organization.	Data breaches, data loss and data damage

(Continued)

Table 1 (continued)

Technical threats	Description of threats	Impact and example of threats
SQL injection attacks [2,3,4,5,9]	Databases are at risk and are vulnerable to SQL injection attacks. An attacker inserts malicious SQL code into user input fields or queries to manipulate the database, potentially resulting in data exposure or unauthorized retrieval. Using SQL injection grants attackers unlimited access to the entire database.	SQL manipulation Code injection Function call injection
DoS attack [1,3,4,5,10]	DoS attack aims to shut down a system and make it unavailable and unreachable (inaccessible) to users by flooding the server with traffic to interrupt its normal functioning. DoS attack is characterized by using a single computer to launch the attack. DoS can be motivated by different factors such as ransom scams or a bug infection.	Protocol based attacks and application layer attacks
Piggybacking attack [9]	A piggyback attack means accessing a system without authorization by exploiting users' credentials.	Unauthorized access
Transmission control protocol (TCP) spoofing attack [9]	Attackers use a spoofed (TCP) connection n to impersonate the Internet Protocol (IP) address trusted by the PostgreSQL server to leak database information by sending a database leaking payload.	Spoofing attacks
TCP Hijacking/injection [9]	TCP injection attacks against existing TCP streams as earlier TCP/IP implementations that had global acknowledgement (ACK) limits could be abused to leak the sequence number of existing connections.	Hijacking attacks
Differential attack [3]	An attack that uses both the plaintext and its cipher text to discover the key that was used to encrypt the plaintext and aims to exploit differences between hashes resulting from a slight change in the input.	Differential attacks on

(Continued)

Table 1 (continued)

Technical threats	Description of threats	Impact and example of threats
Vulnerable audit trail [1,3,4,7]	A shaky database audit policy poses a serious threat to the company, like regulatory danger, deterrence, detection and recovery, lack of user accountability, performance degradation, and separation of duties, limited granularity, and proprietary. Most audit systems don't know who the actual end user is because they are logged under an account name that may not be their real name; this makes it hard to know who did what causing a threat.	Vulnerable audit trail attacks
Exploitation of vulnerabilities [3]	Leaving vulnerabilities unfixed can increase the possibility of threats happening.	Any type of attack can exploit the unfixed vulnerabilities
Advanced persistent threats (APTs) [3]	Highly skilled and targeted attacks that aim to sustain long-term, unauthorized access to databases. APTs frequently combine social engineering techniques, sophisticated malware, and persistent monitoring to gather information and conduct malicious activities.	Social engineering techniques, sophisticated malware, and persistent monitoring to gather information and conduct malicious activities
Excessive privileges [1,4]	Granting excessive permissions to users or not revoking the privileges of ex-employees can sometimes be a threat. Users may abuse privileges for malicious purposes. This threat is one of the most dangerous threats.	User privileges attacks
Database misconfiguration [1,3]	Un-patched databases are targeted by attackers frequently. Unfortunately, organizations struggle to maintain database configurations even when patches are available, because of high workloads, the difficult and lengthy requirements for patch testing, and the difficulty in scheduling a maintenance period to access and fix what is often seen as a business-critical system.	Database misconfiguration attacks
Backup exposure [1,4]	If backup storage devices, such as disks and tapes, are not kept in a secure location and are not monitored, private information may be stolen. This threat has been the focus of many high-profile security breaches.	Backup exposure

(Continued)

Table 1 (continued)

Technical threats	Description of threats	Impact and example of threats
Data tampering [1]	Replacing original data and altering it illegally causes its loss.	Data modification
Malicious traffic [1]	A large number of requests are forged by external attackers to prevent normal users from accessing the database.	DDoS attacks

SQL Injection represents another type of threat, involving the insertion of malicious code into original SQL database queries with the intent to subvert the application's purpose [11]. Poorly written web application code or configuration errors can lead to compromises in integrity, availability, and confidentiality through SQL Injection [12]. Two primary forms of input injection exist: (1) SQL Injection, which targets traditional databases by inserting unauthorized SQL statements into input fields, and (2) NoSQL Injection, which focuses on big data platforms and involves inserting malicious statements into big data components. A successful attack may grant the attacker unrestricted access to the entire database. Furthermore, SQL injection attacks can be conducted through three methods: (1) SQL Manipulation, which alters SQL commands within the application, (2) Code Injection, which exploits computer bugs caused by invalid data processing to add additional SQL statements or commands to existing SQL statements, and (3) Function Call Injection, which inserts database or operating system function calls into vulnerable SQL statements to manipulate data or execute privileged system calls [13,14]. Another technical threat involves graphical passwords, which are user-friendly authentication mechanisms. However, their increasing popularity correlates with a rise in their vulnerability to security attacks. As outlined in Table 1, security attacks on passwords can be categorized into twelve distinct types:

1. Shoulder surfing:

This form of attack frequently occurs in crowded environments. It involves a malicious actor observing an individual as they input a password on a computer, with the intent of gaining unauthorized access to private or sensitive information.

2. Video-recording attack:

Utilizing a device equipped with a camera, a malicious actor can record the login process and subsequently replay the video to extract the password [15].

3. Smudge attack:

When users input their passwords by tapping on touchscreen devices or drawing patterns, residual oils and dirt from their fingers leave smudges on the screen. These smudges can potentially be analyzed by attackers to deduce the password [16].

4. Spyware attacks:

Malicious software installed by an attacker on a victim's device can record information and actions. One example is a screen scraper, which captures user activity displayed on the screen [17].

5. Brute-force attack:

A brute-force attack involves an attacker attempting to guess a password by systematically trying every possible combination until the correct one is identified. This method typically targets passwords with a limited character space. The attack may utilize a collection of forge-fingerprint details or Optical Character Recognition (OCR) techniques in the case of CAPTCHAs (Completely Automated Public Turing test to tell Computers and Humans Apart).

6. Computer vision attack:

This method employs artificial intelligence to ascertain an individual's password by analyzing a live or pre-recorded video of the user's finger interactions with a touch screen. The video footage is input into a system that tracks finger movements from the camera's perspective. Utilizing sophisticated algorithms, the system generates multiple patterns, which are subsequently transformed from the camera's viewpoint to the user's perspective. These patterns are then ranked according to predefined criteria. Ultimately, the system presents the attacker with potential graphical password(s), potentially facilitating unauthorized access.

7. Dictionary attack:

A dictionary attack is a password-cracking technique that employs a systematic key approach. In this method, the attacker systematically attempts all possible passwords from a precompiled list, which is typically based on common user behaviors and patterns.

8. Eavesdropping:

A man-in-the-middle attack occurs when an attacker intercepts communication between the user and the server. The attacker either decrypts information sent by the user to the server or intercepts the user's request and replays it to the server. Data eavesdropping, also known as sniffing or snooping, involves the interception of data when an attacker exploits insecure or vulnerable networks to read or steal information as it travels between two devices during the transmission process.

9. The frequency of occurrence analysis (FOA) attack:

This method analyzes the frequency of recurring patterns during the login process. It identifies and restricts the keys utilized for authentication, and depending on the scheme's design, it examines either image or keypress location frequency.

Image frequency analysis refers to a method employed by malicious actors to deduce passwords by identifying and exploiting patterns in the most frequently occurring images.

Keypress location frequency analysis: determining the most frequently selected final image location for authentication by generating a heat map based on occurrence data.

10. Social engineering attacks:

Persuading users to unknowingly divulge their information. Phishing represents a prominent social engineering tactic.

11. Image gallery attacks:

Physical access attacks occur when an unauthorized individual gains direct access to a physical server or database. This type of breach potentially enables the attacker to log in as any user, circumvent authentication protocols, and alter images utilized during the authentication process.

12. Sonar attack:

In a sonar attack, an adversary detects the user's device-unlocking gesture and infers the pattern utilized by analyzing recorded sound waves. The attack exploits the device's microphone to capture a frequency emitted by the application, which is often imperceptible to human hearing.

4.1.2 Classification of Non-Technical Threats

Non-technical threats refer to dangers arising from human activities that may lead to unauthorized access to sensitive data. In this study, these threats are classified based on their nature and sources, as illustrated in Table 2. Non-technical threats encompass human vulnerabilities that can be exploited to breach database systems. These can be categorized into insider threats, human errors, physical thefts, third-party risks, social engineering, data exposure, data tampering, illegal user behavior, and unauthorized access.

Table 2: Classification of non-technical threats in database systems

Technical threats	Description of threats	Impact and example of threats
Human error [6,7]	Human mistakes such as accidental disclosures of sensitive information, misdirected emails, and unintentional disclosure of login credentials.	Disclosures of sensitive information, misdirected emails, unintentional disclosure of login credentials
Insider threats [3,4,7]	People who have access to sensitive data can intentionally misuse this data for malicious reasons. They are considered as threatening as the outsider threats. Insiders may be disgruntled employees, contractors and business partners. Some tactics that malicious users may use are copying files onto a Universal Serial Bus (USB) drive, emailing sensitive information to a personal account, sharing access credentials with unauthorized individuals, or even planting malware or other hacking tools to facilitate their activities. Insider threats can also be inadvertent negligence by individuals.	Internal threats from employees, lack of awareness and employee negligence
Third-party risks [1,4,7]	Some organizations rely on third-party systems or services to manage their data, security vulnerabilities in these third-party systems, or services put enterprise data at risk.	Third-party risks

(Continued)

Table 2 (continued)

Technical threats	Description of threats	Impact and example of threats
Third-party privilege elevation [4]	A villain third-party developer can disable audit mechanisms and take advantage of their authorization to cause harm.	Third-party privilege elevation
Physical theft of assets [7]	Keeping physical assets in insecure places, like leaving them in a room with no safety door or security cameras.	Physical theft of IT devices, network devices
Illegal user behaviors [1]	Violates the role behavior rules in the database such as users' illegal operations in the database system.	Illegal user behaviors
Inaccurate identification [1]	Illegal users are wrongly identified as normal users or the opposite.	Inaccurate identification
Bypass/physical breaches [1]	An attack on database hardware that leads the system to work abnormally happens when the laser fault injector injects faults into the chip supporting the database.	Physical security breaches
Social engineering	Threats exploit people's tendencies to trust them with the goal of stealing sensitive information such as user names and passwords to grant access to the system.	Identity theft, fake business email and social media theft

Fig. 3 illustrates the analysis results of technical and non-technical cyber threats classifications. The findings reveal that SQL injection attacks and DoS attacks were the most prevalent technical threats in database systems, each accounting for 9% of incidents. Vulnerable audit trails, intrusion attempts, and ransomware attacks were identified as the second tier of technical threats, representing 7%, 7%, and 5% of incidents, respectively. The third tier of technical threats comprised phishing, database misconfiguration, attacks on graphical passwords, and backup exposure, each constituting 4% of incidents. Other technical threats, including TCP spoofing attacks, TCP hijacking/injection, piggybacking attacks, and malicious traffic, were observed less frequently, each accounting for 2% of incidents. Regarding non-technical threats, insider threats emerged as the most common, representing 5% of incidents. Other non-technical threats, such as bypass/physical attacks, illegal user behaviors, physical theft of assets, and third-party privilege elevation, were categorized in the second tier, each accounting for 2% of incidents.

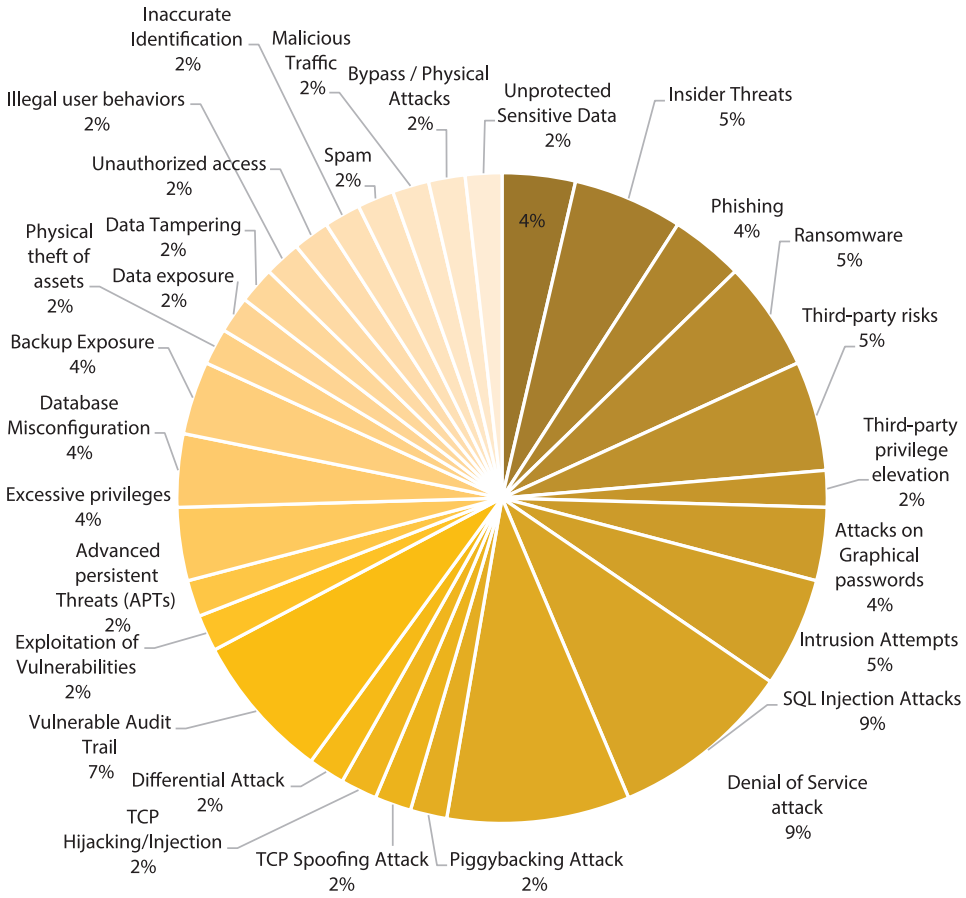

Figure 3: Analysis of classifications of technical and non-technical cyber threats

4.2 Vulnerabilities Identification

The second step of the framework aims to identify the technical security vulnerabilities that could be exploited to compromise the database systems' assets. These vulnerabilities may be associated with either single or multiple operational or cyber security threats. Vulnerability scans and assessments are crucial steps in the risk assessment process to identify critical technical vulnerabilities. In this study, the classification of vulnerabilities is divided into twenty main categories: weak authentication, untrusted third-party, unnecessary third-party granted access, software bugs, unpatched database, insecure coding practices, weak security controls, weak audit trail, limited security expertise and education, unmanaged sensitive data, insecure network, password misuse, platform vulnerabilities, integration challenges, management complexity, weak encryption practices, weak data masking, multiple usage of an account, weak security awareness, and ineffective key management, as shown in Fig. 4. Table 3 summarizes the main technical vulnerabilities in database systems, describing these vulnerabilities and their impacts.

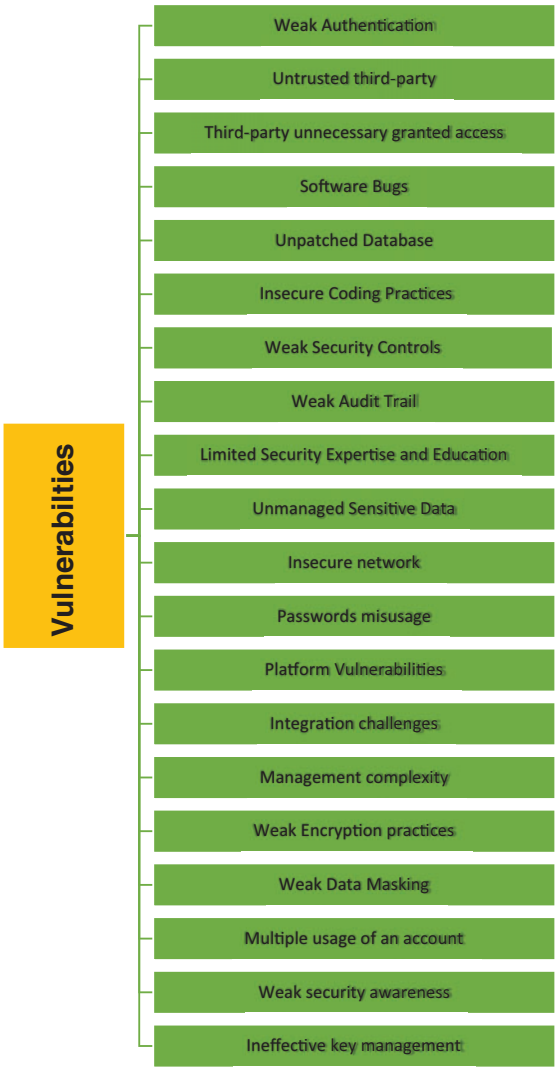

Figure 4: The most common technical security vulnerabilities in database systems

Table 3: Classification of technical vulnerabilities in database systems

Vulnerabilities	Description and impact
Weak authentication [6,7]	Users using weak authentication leads to attackers easily guessing the credentials. Lack of proper training or understanding of company policies can lead to catastrophic loss to an organization.
Untrusted third-party [7]	Third-party systems may not have the same level of security as the enterprise's internal systems, and they may not be updated or maintained regularly.

(Continued)

Table 3 (continued)

Vulnerabilities	Description and impact
Third-party unnecessary granted access [4]	Granting unnecessary access to a third-party admin to a database allows the admin to use this vulnerability to harm the system by modifying, stealing data, or even shutting the system down.
Software bugs [5]	A bug is a problem within software that causes it to work in an improper way, which means it might perform functions while it is not supposed to or not perform other functions it is supposed to.
Unpatched database [1,5]	An incorrect configuration of a system puts it at risk by exposing the system's sensitive data and code. As organizations take months to patch databases, the attackers have more time to exploit the vulnerabilities and launch their attacks.
Insecure coding practices [3,5]	A vulnerability in databases with a lack of secure coding practices makes the system insecure, causing breaches and loss of data and information. Improper implementation or management of encryption can result in vulnerabilities that can expose data to attackers.
Weak security controls [5]	Security threats are malicious attacks that affect a system and cause harm to it. In databases, a threat involves SQL and non-SQL to inject malicious code into a system.
Weak audit trail [1,3]	Main vulnerabilities of audit trail: 1. Information overload: large batches of data can lead to overwhelming security terms making it difficult to identify. 2. False positive: lead to alert fatigue and potential oversight of critical incidents. 3. High costs: implementing and maintaining auditing and monitoring systems can be expensive, involving costs for tools and storage.
Limited security expertise and education [1]	Many organizations struggle to keep up with data growth and are prepared for security breaches. The lack of expertise is the main cause, so they don't handle incidents properly.
Unmanaged sensitive data [1]	Sensitive data in databases will be exposed to threats if the required controls and permissions are not implemented.
Insecure network [3]	Passwords transmitted over insecure networks can be intercepted by hackers.

(Continued)

Table 3 (continued)

Vulnerabilities	Description and impact
Passwords misusage [3]	Using weak passwords or using the same password for many accounts allows the attacker to guess the password easily and access all accounts that use it.
Platform vulnerabilities [4]	Unauthorized entry, data corruption, or service denial can result from flaws in underlying working frameworks and extra services installed on a database server.
Integration challenges [3]	Complex integration means that integrating monitoring tools with existing systems can create gaps in coverage.
Management complexity [3]	Configuration challenges because complex setup and management can lead to ineffective threat detection.
Weak encryption practices [3]	Poor key management can compromise encryption. If encryption keys are not securely stored, rotated, or managed, encrypted data can become vulnerable to unauthorized access.
Weak data masking [3]	Poorly implemented data masking can lead to decreased protection of sensitive data and complexity of permission management to the overall access control system.
Multiple usage of an account [1,3]	When numerous people use the same account, it's more difficult to determine who is responsible for any given action.
Weak security awareness [1]	Database users make the database vulnerable by creating attack points that may be exploited by attackers such as setting weak password and not modifying the default password.
Ineffective key management [3]	Poor key management can compromise encryption. If encryption keys are not securely stored, rotated, or managed, encrypted data can become vulnerable to unauthorized access.

Fig. 5 illustrates the analysis results of technical security vulnerability classifications. The findings indicate that weak authentication, unpatched databases, weak audit trails, and multiple usage of a single account were the most prevalent technical vulnerabilities in database systems, each accounting for 9% of the identified issues. Furthermore, software bugs, insecure coding practices, weak security controls, insecure networks, password misuse, weak encryption practices, and inadequate data masking were classified as secondary-level security vulnerabilities in database systems, each representing 4% of the identified vulnerabilities. Additionally, the study revealed that weak security awareness, limited security expertise and education, untrusted third parties, unnecessary third-party access, and management complexity were the most common non-technical vulnerabilities in database systems, each comprising 4% of the identified issues.

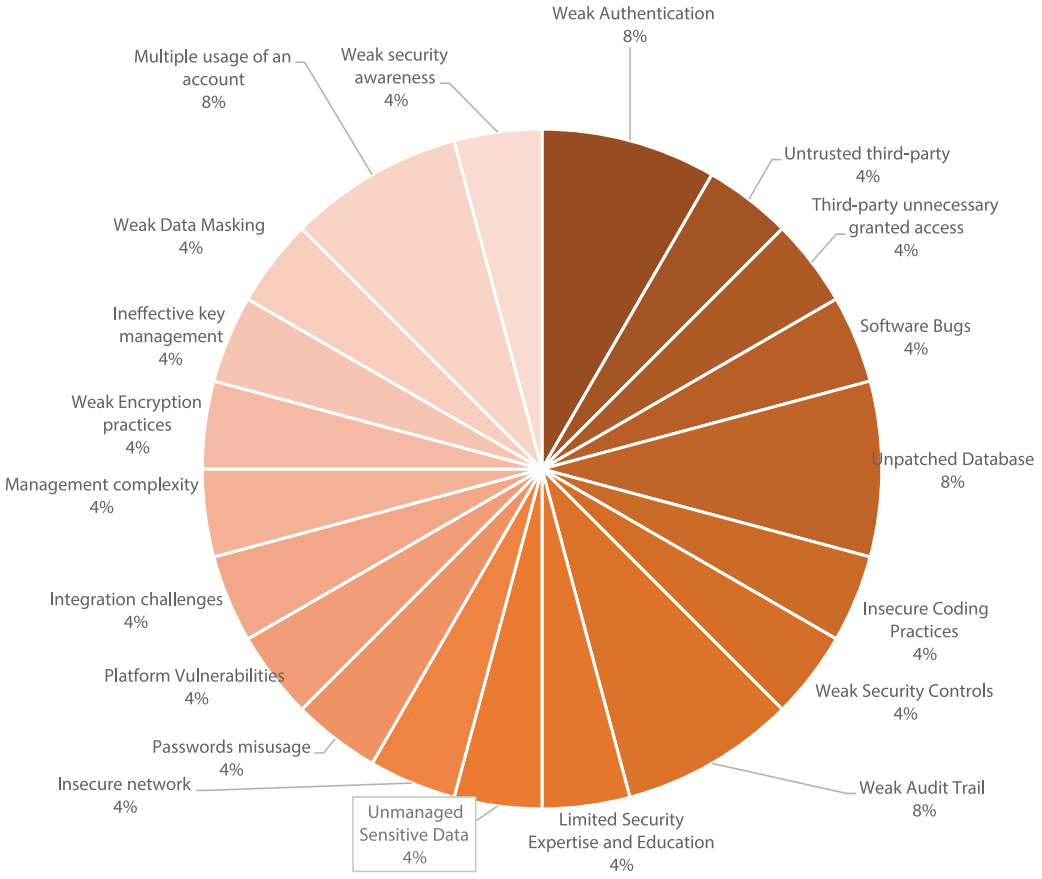

Figure 5: Analysis of classifications of technical security vulnerabilities

4.3 Countermeasures Identification

In this phase, we conduct a comprehensive analysis of essential countermeasures aimed at reducing and mitigating the impact of vulnerabilities associated with cyber threats. Our study identified a range of security controls designed to enhance database system security against cyber-attacks. These measures include data encryption, access control, authentication, firewalls, data backup, behavior detection, spam detection, security audits, anomaly detection methods, and others, as illustrated in Table 4.

Table 4: Classification of countermeasures in database systems

Countermeasures	Description
Data encryption [1,4,5,7]	Encryption is the process of transforming data into a coded format to make it unreadable by intruders and difficult to decipher, whether it is during transmission or at rest. It can be applied to many data types, such as emails, files, databases, and other communication channels. Encryption can also prevent insider threats, as insiders who have access to the data will not be able to read it unless they have the required authorization. In addition, it helps organizations to ensure their confidentiality, integrity, and availability by adhering to several data protection regulations, such as the General Data Protection Regulation (GDPR) and the Health Insurance Portability and Accountability Act (HIPAA). Ensures the security of users by converting the data with the AES algorithm to the database management system, Message Digest (MD5), and Secure Hash Algorithm (SHA-256) to protect network data transmission. It is cost-effective; investing in the implementation of encryption technology is cheaper than dealing with the consequences of data breaches.
Access control [1,2,7,8]	All DBMS use access control to create user accounts and passwords to prevent unauthorized people from entering the database system and obtaining confidential information. Granting and revoking privileges are methods of enforcing access control. The organization must set policies defined by access control that all contact with the databases must adhere to. It is suggested that web tripwire and login rituals be integrated using Multi-Factor Authentication (MFA). Access control allows organizations to do the following: - Access control allows organizations to implement a layered defense approach to security. - Helps organizations follow protection data regulations. - Prevents insider threats. - Allows organizations to detect and respond to security incidents. Access control systems consist of: - File permissions to create, read, edit, or delete files on the server. - Program permissions are the rights of executing an application program on the server. - Data rights, the rights of retrieving, or updating data in a database. Access control mechanisms: 1. Discretionary Access Control (DAC) 2. Mandatory Access Control (MAC) 3. Role-Based Access Control (RBAC)

(Continued)

Table 4 (continued)

Countermeasures	Description
Authentication [3–5]	Users' authentication must be strong, so attackers' opportunities are low to get legitimate rights from targeted users and then steal or modify credentials in database systems. It is important to use advanced technologies and rules and implement strict usernames and passwords to prevent authentication attacks. Also, using directory integration, which is using a specific login detail for multiple databases and programs, but with a double-factor authentication system which requires two credential categories or a multi-factor authentication system which requires more than two credential categories, increases the scalability and simplicity of use. Classifying users and determining what privileges and access permissions they have is a basic security requirement. Users can be authenticated externally using Operating Systems (OS) or network services. User authentication can be established using Secure Sockets Layer (SSL), business parts, and middle-tier server authentication, also known as proxy authentication. Implementing a secure authentication protocol for data transmission between the server and the client by ensuring that the values sent are randomized and unique can reduce eavesdropping attacks. Some authentication methods are: 1. Biometrics 2. Tokens 3. Multi-factor authentication
Firewalls [5,7,9]	Firewalls monitor databases and defend them against attacks. Using the logs they keep, they can audit and monitor every database access. Firewalls can: - Offer a level of control over network traffic. - Prevent unauthorized access to sensitive data. - Allow organizations to monitor and control their employees' activities on the Internet. - Allow organizations to identify and block advanced threats.
Data backup [1,4,7]	Backup of databases requires an encryption application. Backups offer the following: - Provides a method to recover lost or compromised data. - Helps avoid the loss of data due to accidental deletion. - Helps organizations follow data protection regulations. - Protects against insider threats.
Behaviors detection [1]	Real-time tracking and analysis of database operation information and data models such as naive Bayesian classification algorithm
Spam detection [1]	A spam classification model, which is a machine learning method, is used to improve spam detection.
Security audit [1]	Testing of the organization's information systems to assess the security of it.
Anomaly detection [1]	A method to detect bypass attacks based on virtual machine cache by monitoring technology and hardware performance.

(Continued)

Table 4 (continued)

Countermeasures	Description
Emerging materials [1]	Materials such as silicon nanowire field-effect transistors and nanoelectromechanical switches have security advantages over traditional materials.
Training employees [1,7,8,10]	Training and educating employees about the importance of security measures and the latest threats through awareness programs can limit social engineering attacks like phishing. Also, educating employees about covering their fingers while drawing a pattern or selecting characters can reduce computer vision attacks and video recording attacks. Security awareness allows organizations to reduce the number of human-related incidents and help combat human errors.
IT security expertise [8]	Employ Information Technology (IT) security professionals to regularly perform vulnerability scans and penetration testing and find suitable controls for risks that may occur.
Security cameras [8]	Implementing security cameras helps prevent unauthorized access to physical areas, reducing the risk of physical threats and sabotage.
Network segmentation [8]	Aids in restricting the spread of cyberattacks throughout the organization's network and isolating vital resources and assets.
Intrusion detection and prevention systems (IDPS) [1,2,9,10]	Designed to detect and respond to new and advanced attacks. IDS evaluates data using network traffic, database operations, SQL queries, system logs, etc. When an attack is identified, Intrusion Prevention Systems (IPS) stop them by either disabling connections, blacklisting IP addresses, or changing firewall settings. IDPS combines signature-based and behavioral detection approaches. These two approaches help identify zero-day attacks. An example of IDS is Host-based Intrusion Detection Systems (HIDS). An IDS can detect SQL injection attacks by monitoring the traffic on the database system for patterns, features that are linked with SQL injection attacks, malicious activity, and policy violations.
Incident response plans [7]	They include procedures for reducing the negative effects of cyberattacks and communication strategies for alerting stakeholders in case of a data breach.
Biometric authentication methods [1,7]	An effective linear binary pattern called Fourier transform is used to process and store the biometrics of hand type and iris of users into the database. They enhance the safety of physical access to sensitive areas. Examples are fingerprint, facial, and iris recognition.
Anti-phishing [8]	Software and tool-based strategies for phishing attack mitigation. These compromise stand-alone systems, methods for designing programs, and tools for mitigating purposes.
Models and frameworks [8]	Models and frameworks help mitigate phishing attacks. It includes frameworks that regulate a series of activities and machine learning-based models and methods to improve the anti-phishing capabilities of newer or existing systems.

(Continued)

Table 4 (continued)

Countermeasures	Description
Human-centric mitigation strategies [8]	These techniques impact human users' ability to recognize and mitigate phishing attempts more effectively. They mostly include guidelines for enhancing these skills, such as planning and carrying out anti-phishing training, conducting evaluation quizzes, etc.
Obfuscation [10]	It is a technique used to reduce shoulder surfing attacks by making authentication information unclear to onlookers. One technique is hiding or decoding the real input during authentication. For example, hiding password components among decoy images makes it challenging for the onlooker to identify the correct selection. There are two methods for obfuscation: 1. Graphical One-Time Password (GOTPass) 2. EvoPass
Randomization [10]	It is a technique used to reduce shoulder surfing attacks and brute-force attacks by randomizing the arrangements or positions of password elements. There are two methods for randomization: 1. Coin Passcode Model 2. 2D Coordinates System Randomization can also reduce smudge attacks, computer vision attacks, and guessing password attacks. Guessing password attacks are randomized using the Click-based Captcha as a Graphical Password (CaRP) technique.
RiS and T-RiS [10]	Rotating into Sector (RiS) and Rotating into Sector Based on Text (T-RiS) were developed to incorporate randomness or visual complexity to confuse potential attackers who try to capture passwords through video recording. Both feature a Long Reach 1 (LR1) login mode that improves security by introducing three concentric rings. The user alone can discern the location of the line or sector, and it changes randomly after each character entry to add a visual challenge for attackers attempting a video recording attack.
Dynamic screen changes [10]	Devices repeatedly change the screen's color and brightness to confuse the camera of the attacker who is recording the authentication process. This reduces computer vision attacks.
Performing test during authentication [10]	Completely Automated Public Turing Tests to Tell Computers and Humans Apart (CAPTCHA) involves conducting tests during authentication. The common goal is to generate problems that are beyond the capabilities of computer programs and can only be solved by humans. This technique limits spyware attacks.
Large password space [10]	An example of Large Password Space: The Vibration-and-Pattern (VAP) code.

(Continued)

Table 4 (continued)

Countermeasures	Description
Additional on-screen activities [10]	Incorporating additional on-screen activities during the authentication process improves security. For example, using a Swype-like method or sketching various graphical shapes before or after drawing the pattern makes it more difficult for attackers as the authentication process becomes more complex. This reduces computer vision-based attacks.
Skipping dots [10]	A user intentionally skipping dots while drawing a pattern makes it harder for the algorithms to identify which dots have been skipped. This reduces computer vision-based attacks.
Conundrum-pass [10]	This approach starts by asking users to choose an image and choose a number, n. The chosen image is divided into an n × n square matrix. Then, users can choose specific image chunks to form the patterns they desire. During the login session, the images are randomly shuffled and organized. The users must choose the previously selected grid in the correct order. This method combines randomness and complexity and adds it to the pattern selection process, which makes it more resistant to dictionary attacks.
Spin-wheel-based authentication [10]	This approach includes a spin-wheel-based graphical authentication mechanism. It provides a large password space. Users are given a spin wheel containing four sub-wheels each with 36 slots filled with randomly arranged numbers from 1 to 36. Users must choose a number from each sub-wheel and arrange the numbers in a row by spinning the wheel to create a password for authentication. This method combines both complexity and unpredictability into the password creation process, which makes it more resistant to dictionary attacks.
Hashing timestamps and pass-image components [10]	A security measure suggested by English and Poet. It requires applying hashing to several authentication components, such as timestamps pass-image related data, to make it difficult for attackers to decipher the authentication data which appears as random and unintelligible strings, this makes it resistant to eavesdropping.
Random location assignment for passphrase [10]	A security measure suggested by English and Poet. For each authentication attempt, the location of the passphrase is assigned. For every attempt, the passphrase is different and is sent to the server for verification ensuring that the data sent is always unique and different, which reduces the risk of eavesdropping attacks.
Countermeasures against FOA attacks based on image frequency [10]	- Use of Decoy Images - Display of "Dummy Screens" on Failed Attempts - Limit on Failed Authentication Attempts

(Continued)

Table 4 (continued)

Countermeasures	Description
Countermeasures against FOA attack based on pass image location [10]	- Dynamic Generation of Target Images - Algorithmic Determination of Target Images - Random Distribution of Target Images
Click text scheme [10]	Within the CaRP system, the "Click Text" scheme uses a random arrangement of alphanumeric and special characters inside an image that poses a challenge. Because of the addition of complexity and unpredictability this randomization offers to the authentication process, it will be more difficult for the attacker to determine the right password. In other words, it reduces the chances of guessing attacks.
Animal grid scheme [10]	Another approach of CaRP, for the authentication process, is that it uses 2D animal images. The animal grid scheme reduces guessing attacks because it adds both randomness and variability to the arrangement of these images.
HTTPS verification [10]	Verifying the presence of a Hypertext Transfer Protocol Secure (HTTPS) prefix before entering a password on a website reduces the chances of phishing attacks, because HTTPS offers a secure and encrypted connection as it employs SSL and Transport Layer Security (TLS) protocols.
Watermarking techniques [10]	They are used to protect against image gallery attacks and unauthorized modification to the images in the gallery. It gives a unique watermark to each digital image using a secret key to specify the location of the image. This reduces the image gallery attack.
Verification using secret key [10]	A user can determine if the image was tampered with or remains untampered by extracting the water mark of an image and comparing it with the given watermark using the secret key. This allows the user or the system to ensure the integrity and authenticity of images and reduces the image gallery attack.
Password hashing methods such as MD5, Bcrypt, Argon2, and Scrypt [6].	Password hashing methods, such as MD5, Bcrypt, Argon2, and Scrypt, have a cryptographic hashing function property such as the avalanche effect.
Braille transformation [6].	Securing passwords using Braille Transformation before storing them in a database as it transforms the hash according to the following steps: 1. Password entry and hash generation 2. Braille transformation 3. Matching with the Braille code 4. Encryption reinforcement 5. Database storage Braille Transformation parameters: - Irreversibility - No collisions

(Continued)

Table 4 (continued)

Countermeasures	Description
Honeypots [1,2]	A honeypot is a deception device used to attract and trap intruders. SQL injection attacks can be uncovered with the use of a honeypot, which acts as a vulnerable web application to attract hackers' attention.
Log analysis [2]	Analyzing log files created by the web server and the database server can help find security holes in a system. A technique to use log analysis to detect SQL injection attacks is to review the log files for indications of malicious code in an SQL query and can be analyzed by hand or with tools.
Signature-based detection [2]	Signature-based detection uses a unique signature, or digital footprint, from software programs running on a protected system. Antivirus programs scan software, identify the signature then compare it to signatures of known malware.
Input validation [2]	The first line of defense against SQL injection attacks is to ensure the input is valid by checking against predefined criteria to ensure that it can be processed by the software. The fundamental goal of input validation is to ensure that no harmful code is delivered to the database as part of an SQL query. Data type validation, range validation, and character set validation are just a few examples of input validation techniques.
Parameterized queries [2]	A prevention technique that queries provide a safe method of executing SQL statements by separating the arguments from the main SQL query. By removing the attacker's ability to insert harmful code into the query, the possibility of a SQL injection attack is eliminated.
Stored procedures [2]	Precompiled SQL statements that are kept in the database provide further protection against SQL injection attacks. Stored procedures make it harder for an attacker to inject malicious code into the SQL query by having it performed on the database server rather than the web server whereas the database server is safer than the web server.
Inference control [5]	It protects the system's statistical database, which is of a higher level of importance. Queries only target statistics to protect individuals' data such as SUM, AVERAGE, and MAX. Inference control helps prevent information from getting revealed indirectly. Unauthorized data disclosure can occur in 3 ways: 1. Correlated data: There is a semantic link between visible data X and invisible data Y in popular channels. 2. Missing data: When NULL values in the query mask sensitive data, that way, existing data could be detected. 3. Statistical inference: this is common in databases that hold numerical information regarding individuals.
Flow control [5]	Examines and monitors how information flows through a transaction or program. It is used to stop the flow of requests by unauthorized users who want to access detailed confidential information.

(Continued)

Table 4 (continued)

Countermeasures	Description
XML control [5]	Extensible Markup Language (XML) security standards such as digital signature and encryption. Syntax and processing specification of XML signature describes an XML syntax for representing the association between cryptographic signatures and XML documents.
Digital signatures [5]	Digital Signatures use encryption to provide authentication services in transactions. The goal of it is to link a unique user with a particular text.
Digital certificates [5,10]	A Digital Certificate is a digitally signed statement that combines a public key value with the identity of a service or person that holds the corresponding private key and is issued by a Certification Authority (CA). Certificates lower the chance of social engineering attacks.
IP-based authentication [9]	Feedback channels reduce IP spoofing costs during the handshake spoofing phase, increasing the attack cost. Feedback channels consist of some app-specific feedback channels: - Simple Mail Transfer Protocol (SMTP)-specific feedback - DNS-based feedback - Email-based feedback - Local end-to-end experiment - Real-world experiment
Source address validation (SAV) [9]	Mitigating IP spoofing by performing an outbound Source Address Validation (SAV) to drop traffic that spoofs IP addresses outside of their announced prefixes.
SMTP synchronization [9]	After establishing a TCP connection with an SMTP server, an SMTP client should only send the HELO message after receiving the greeting message from the server. As for plain TCP connections, the send mail team rejects SMTP clients that send further payloads without waiting for a reply. Forcing clients to wait for the servers' reply to messages can stop the adversary from adding the payload to trigger feedback channels.
STARTTLS [9]	STARTTLS provides opportunistic TLS, which offers a way to upgrade a plain text connection to an encrypted connection. Enforcing the use of STARTTLS will stop TCP spoofing and thus eliminate the motivation for feedback channels.
TCP/IP stack disclosure [9]	Ghost ACK packets acknowledging data that is never sent by the attacker can be dropped by the TCP/IP stack. Information Services Network (ISN) is stored for established connection until the sequence number space wrap-around, so that the server can verify and drop ACK packets acknowledging sequence number lower than the ISN (ghost ACKs).
Query-level access control [4]	Through a process called question-degree, access to the database is limited to the bare minimum of SQL operations (select, create, etc.) and facts.
Legitimate privilege abuse prevention [4]	Using a policy of patron packages, place, and time, database access managers can know who is responsible for misusing the database. It patches the privilege abuse threat.

(Continued)

Table 4 (continued)

Countermeasures	Description
Privilege elevation preventive (IPS & QLAC) [4]	It uses a combination of traditional Institution Prevention Systems (IPS) and Query Level Access Control (QLAC) to prevent the exploitation of privilege raise. IPS examines databases to ensure there are no patterns that may lead to weaknesses. However, if a weakness is found, the IPS blocks the entire access to the prone method or uses embedded attacks to block the most successful process.
Preventing weak audit [1,4]	Most of the vulnerabilities that exist with the local audit equipment get fixed by good quality network-based audit home equipment. - High Performance - Separation of Duties - Cross-Platform Auditing
DoS prevention [4]	DoS attack prevention requires multiple layers of protection. Network, software, and database layers are all necessary. This study focuses on database-specific security. The recommendations focus on query access control, IPS, and reaction timing controls in database-specific contexts.
Audit and accountability [1,4]	Audit to monitor the database activities and make it work on track to ensure the integrity of the data. We may need a third party that makes the auditing that could read the data. Auditing is monitoring and recording several activities within the database systems. Audit logs offer a chain of evidence for tracking, investigating, and detecting security breaches or questionable behavior.
Data masking [3]	Data masking is a way to conceal sensitive data in a database by inserting factious or altered data to change it. It uses methodologies like encoding, character scrambling, and data substitution. This countermeasure safeguards data better during the process of product creation, testing, and analysis. In non-production situations, it balances usability and security.
Tokenization [3]	Tokenization is a way to safeguard sensitive information by substituting real data with fictitious tokens. It negates the need to retain or transmit sensitive data in its original form, which reduces the chance of data breach. It is commonly used in the payment card business.
Hardware security models (HSMs) [3]	To prevent cryptographic keys from falling into the wrong hands, keys are secured and stored in a hardware security model. Using an encryption algorithm and strong user authentication, data is stored in software or a database, and its integrity is maintained. When a key is not needed, it should be well destroyed. Using the key escrow system allows keys to be stored in a safe place and helps recover the key even if the key is lost due to system failure.

Security controls and countermeasures are mechanisms and tools developed to protect database systems from cyber threats and attacks. These countermeasures are crucial for maintaining data integrity and safeguarding database systems from unauthorized access. They can be categorized into several types based on their functions, usage, effectiveness, and importance. As shown in Fig. 6a,b, encryption methods are considered one of the most powerful technical security controls for database

systems, protecting sensitive data and preventing unauthorized access. Multi-factor authentication is also a robust technique for preventing unauthorized access to sensitive data and database systems. Firewalls offer monitoring capabilities for databases and defend them against attacks. Using the logs they maintain, they can audit and monitor all database access. Firewalls provide a level of control over network traffic and prevent unauthorized access to sensitive data. Network segmentation helps restrict the spread of cyberattacks throughout the network and isolate vital resources and assets. IDPS is designed to detect and respond to new and advanced attacks. IDS evaluates data using network traffic, database operations, SQL queries, system logs, and other sources. When an attack is identified, IPS stop them by disabling connections, blacklisting IP addresses, or modifying firewall settings. IDPS combines signature-based and behavioral detection approaches, which help identify zero-day attacks. Fig. 7 represents the most common countermeasures in database systems.

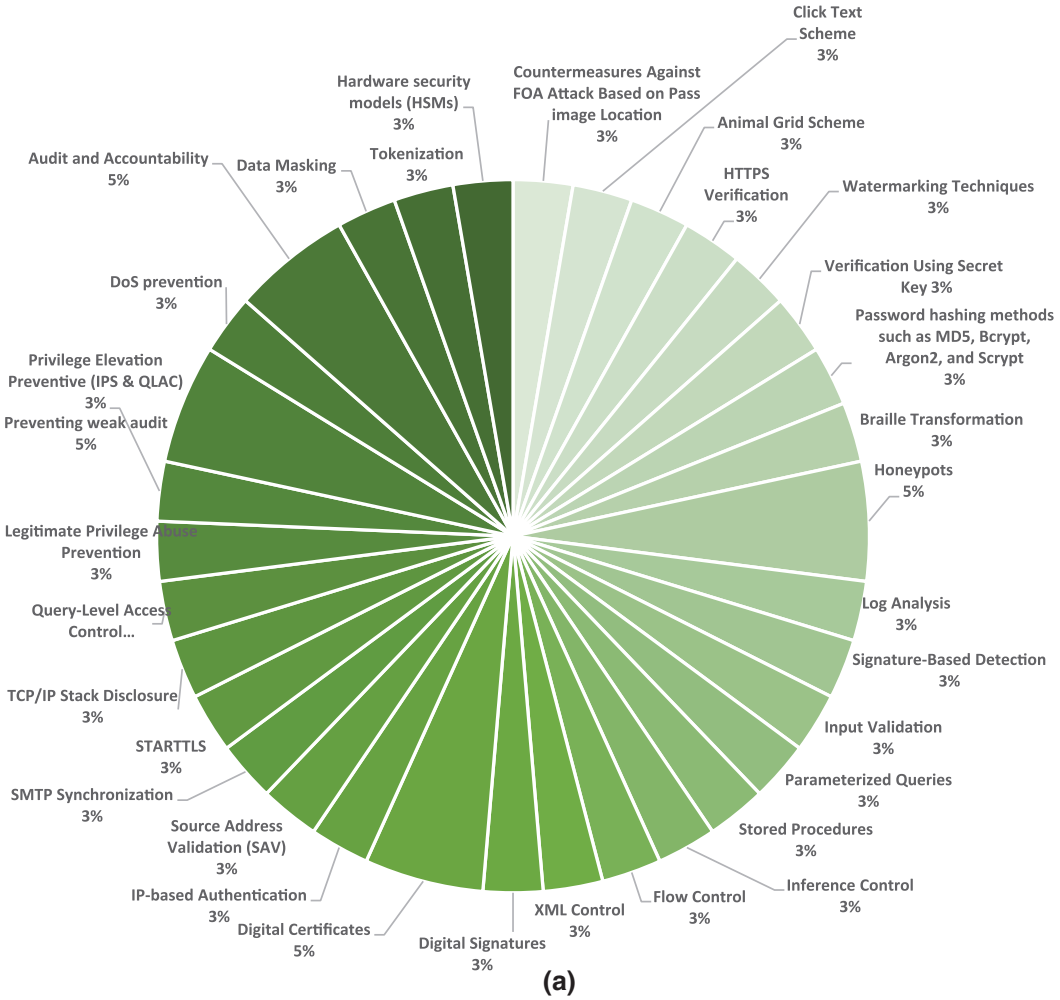

Figure 6: (Continued)

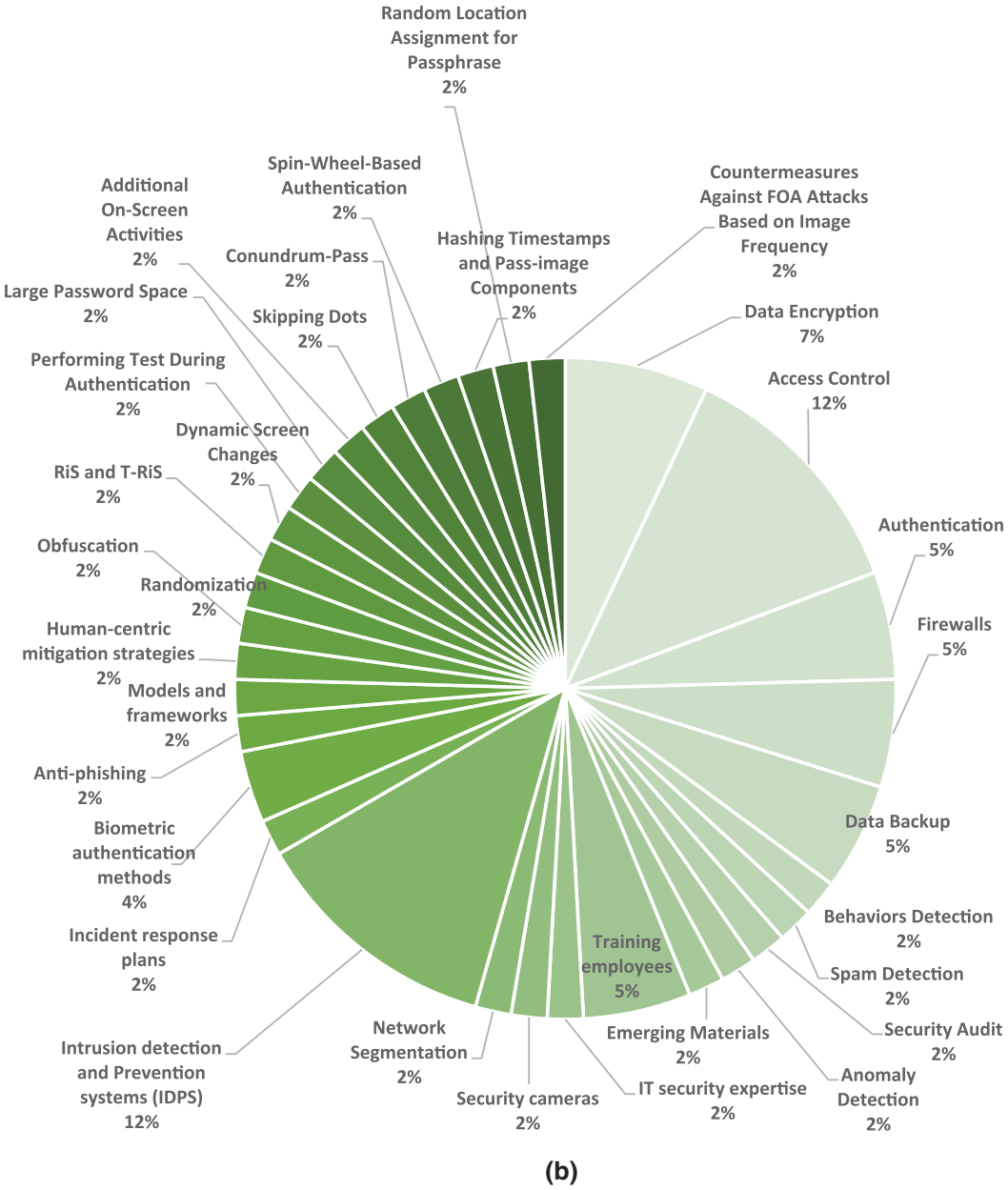

Figure 6: Analysis of classifications of technical security countermeasures

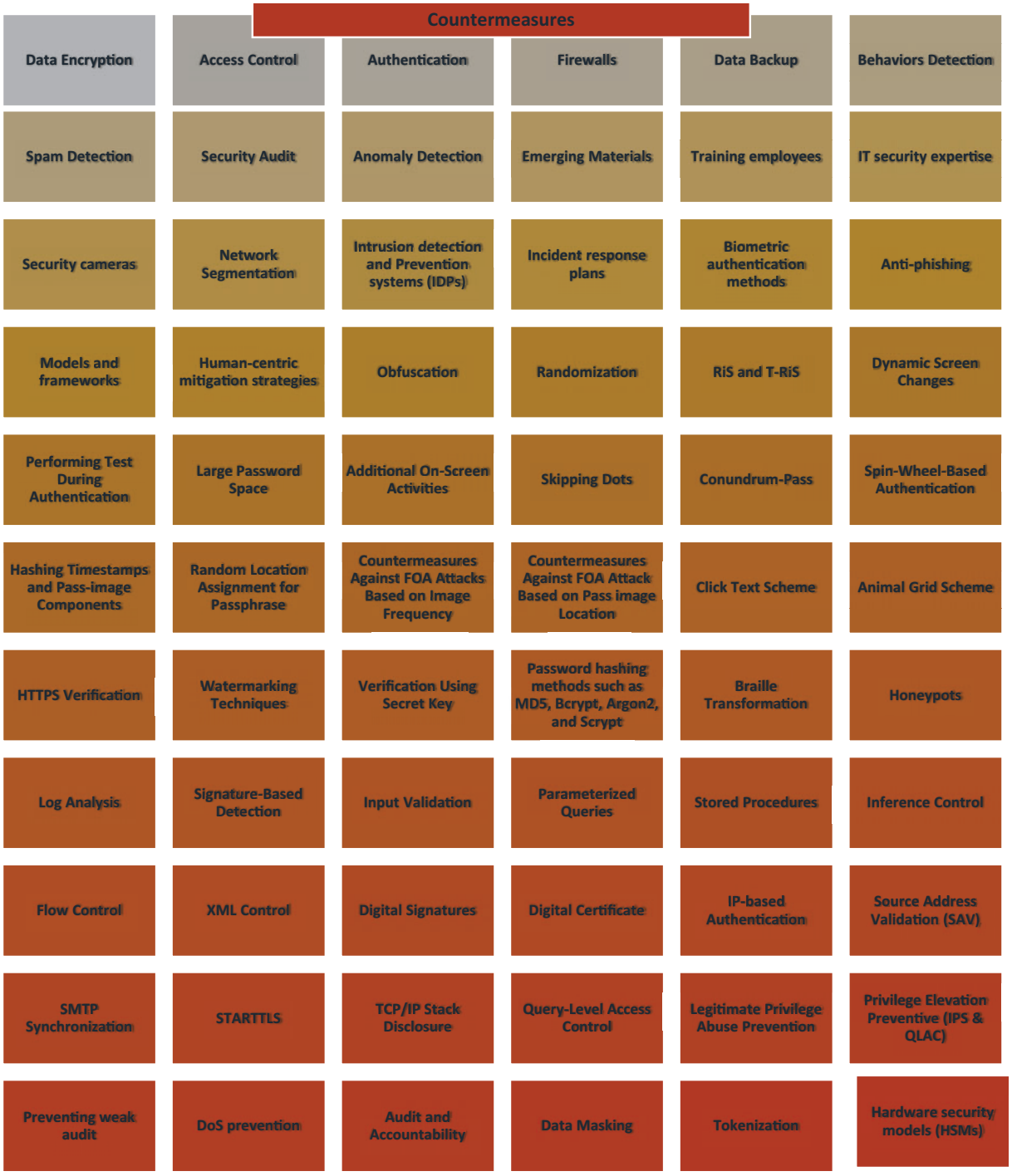

Figure 7: The most common countermeasures in the database systems

5 Research Limitations

While this research offers a comprehensive analysis of cyber risks in database systems, including the classification of threats, vulnerabilities, impacts, and countermeasures, certain limitations warrant consideration. Firstly, cybersecurity threats and vulnerabilities are in constant evolution, with new threats emerging continuously. Consequently, the identification and classification of these threats may become outdated. Organizations, scholars, and researchers must therefore remain vigilant in updating their threat investigations to stay ahead of malicious actors. Secondly, cyber threats are often interconnected and may occur simultaneously or in rapid succession. For example, a cybercriminal might initiate a phishing attack to breach database systems, subsequently deploying ransomware once access is gained. In such scenarios, traditional methods of addressing threats in isolation may prove insufficient. Thus, a more holistic approach to threat management is necessary. Lastly, cyber threat classifications typically consider various factors, including attack vectors, attacker motivations, and organizational impact. However, these factors are often complex, as cyber-attacks can have multiple, overlapping motivations and consequences. An attack on a database system, for instance, might be driven by financial gain, political objectives, or personal vendetta. This complexity challenges accurate identification and classification of attacker intentions, necessitating organizational vigilance and adaptability in response. While cyber risk classifications are crucial for effective defense in database systems, organizations, scholars, and researchers must also recognize the associated limitations and challenges.

6 Conclusion

In the current digital landscape, cyber threats pose the most critical challenges to database management systems, with an increasing number of sophisticated cyber-attacks targeting these systems. The impact of cyber risks on database management systems can be severe, potentially resulting in data loss, reputational damage, and system failure. Consequently, it is crucial to comprehend the behavior of cyber threats on database systems and identify appropriate countermeasures to mitigate their effects. Cyber risk classification and assessment play a vital role in risk management, establishing a significant framework for identifying and responding to cyber threats. Risk assessment facilitates understanding the impact of cyber threats and developing suitable security controls for risk mitigation. This study presents a comprehensive analysis of cyber risks in database management systems, including the classification of threats, vulnerabilities, impacts, and countermeasures. This classification aids in understanding the appropriate security controls required to mitigate cyber risks for each type of threat.

The study's findings revealed that SQL injection attacks and DoS attacks were the most prevalent technical threats to database systems, each accounting for 9% of observed threats. Vulnerable audit trails, intrusion attempts, and ransomware attacks were identified as the second tier of technical threats, representing 7%, 7%, and 5% of threats, respectively. Additionally, insider threats emerged as the most common non-technical threat to database systems, constituting 5% of observed threats. Furthermore, the results indicated that weak authentication, unpatched databases, weak audit trails, and multiple usage of a single account were the most frequent technical vulnerabilities in database systems, each accounting for 9% of observed vulnerabilities. The second tier of security vulnerabilities included software bugs, insecure coding practices, weak security controls, insecure networks, password misuse, weak encryption practices, and inadequate data masking, each representing 4% of observed vulnerabilities.

The classification framework presented in this study serves as a vital tool for practitioners, policymakers, and researchers to identify, classify, and mitigate cyber threats within database systems.

This research provides a comprehensive analysis of the classification of cyber threats, vulnerabilities, impacts, and countermeasures in database systems. The findings from this work can assist organizations in understanding the types of cyber threats and developing robust strategies against cyber-attacks.

Acknowledgement: Not applicable.

Funding Statement: This work was supported by the Deanship of Scientific Research, Vice Presidency for Graduate Studies and Scientific Research, King Faisal University, Saudi Arabia (Grant No. KFU242068).

Author Contributions: Study conception and design: Mohammed Amin Almaiah, Leen Mohammad Saqr, Leen Ahmad Al-Rawwash, Layan Ahmed Altellawi, Romel Al-Ali, Omar Almomani; data collection: Mohammed Amin Almaiah, Leen Mohammad Saqr, Leen Ahmad Al-Rawwash, Layan Ahmed Altellawi, Romel Al-Ali, Omar Almomani; analysis and interpretation of results: Mohammed Amin Almaiah, Leen Mohammad Saqr, Leen Ahmad Al-Rawwash, Layan Ahmed Altellawi, Romel Al-Ali, Omar Almomani; draft manuscript preparation: Mohammed Amin Almaiah, Leen Mohammad Saqr, Leen Ahmad Al-Rawwash, Layan Ahmed Altellawi, Romel Al-Ali, Omar Almomani. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: Not applicable.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] Y. Wang, J. Xi, and T. Cheng, "The overview of database security threats' solutions: Traditional and machine learning," *J. Inf. Secur.*, vol. 12, no. 1, pp. 34–45, 2021. doi: 10.4236/jis.2021.121002.

[2] V. Abdullayev and A. S. Chauhan, "SQL injection attack: Quick view," *Mesopotamian J. Cyber Security*, vol. 2023, pp. 30–34, 2023. doi: 10.58496/MJCS/2023/006.

[3] H. Omotunde and M. Ahmed, "A comprehensive review of security measures in database systems: Assessing authentication, access control, and beyond," *Mesopotamian J. Cyber Security*, vol. 2023, pp. 115–133, 2023. doi: 10.58496/MJCSC/2023/016.

[4] R. A. Teimoor, "A review of database security concepts, risks, and problems," *UHD J. Sci. Technol.*, vol. 5, no. 2, pp. 38–46, 2021. doi: 10.21928/uhdjst.v5n2y2021.pp38-46.

[5] X. Pan, A. Obahiaghon, B. Makar, S. Wilson, and C. Beard, "Analysis of database security," *Open Access Library J.*, vol. 11, no. 4, pp. 1–9, 2024. doi: 10.4236/oalib.1111366.

[6] H. Touil, N. El Akkad, K. Satori, N. F. Soliman, and W. El-Shafai, "Efficient braille transformation for secure password hashing," *IEEE Access*, vol. 12, pp. 5212–5221, 2024. doi: 10.1109/ACCESS.2024.3349487.

[7] V. Bandari, "Enterprise data security measures: A comparative review of effectiveness and risks across different industries and organization types," *Int. J. Bus. Intell. Big Data Anal.*, vol. 6, no. 1, pp. 1–11, 2023.

[8] B. Naqvi, K. Perova, A. Farooq, I. Makhdoom, S. Oyedeji and J. Porras, "Mitigation strategies against the phishing attacks: A systematic literature review," *Comput. Secur.*, vol. 132, 2023, Art. no. 103387. doi: 10.1016/j.cose.2023.103387.

[9] Y. Pan and C. Rossow, "TCP spoofing: Reliable payload transmission past the spoofed TCP handshake," in *2024 IEEE Symp. Security and Privacy (SP)*, San Francisco, CA, USA, 2024, pp. 4497–4515.

[10] L. Y. Por, I. O. Ng, Y. L. Chen, J. Yang, and C. S. Ku, "A systematic literature review on the security attacks and countermeasures used in graphical passwords," *IEEE Access*, vol. 12, pp. 53408–53423, 2024. doi: 10.1109/ACCESS.2024.3373662.

[11] H. Ahmad, I. Dharmadasa, F. Ullah, and M. A. Babar, "A review on C3I systems' security: Vulnerabilities, attacks, and countermeasures," *ACM Comput. Surv.*, vol. 55, no. 9, pp. 1–38, 2023. doi: 10.1145/3558001.

[12] S. M. Toapanta, O. A. Quimis, L. E. Gallegos, and M. R. Arellano, "Analysis for the evaluation and security management of a database in a public organization to mitigate cyber-attacks," *IEEE Access*, vol. 8, pp. 169367–169384, 2020. doi: 10.1109/ACCESS.2020.3022746.

[13] E. Altulaihan, M. A. Almaiah, and A. Aljughaiman, "Cybersecurity threats, countermeasures and mitigation techniques on the IoT: Future research directions," *Electronics*, vol. 11, no. 20, 2022, Art. no. 3330. doi: 10.3390/electronics11203330.

[14] Ö. Aslan, S. S. Aktuğ, M. Ozkan-Okay, A. A. Yilmaz, and E. Akin, "A comprehensive review of cyber security vulnerabilities, threats, attacks, and solutions," *Electronics*, vol. 12, no. 6, 2023, Art. no. 1333. doi: 10.3390/electronics12061333.

[15] M. A. Almaiah, "A new scheme for detecting malicious attacks in wireless sensor networks based on blockchain technology," in *Artificial Intelligence and Blockchain for Future Cybersecurity Applications*, Cham, Switzerland: Springer, 2021, pp. 217–234.

[16] R. Kumar and R. Goyal, "On cloud security requirements, threats, vulnerabilities and countermeasures: A survey," *Comput. Sci. Rev.*, vol. 33, pp. 1–48, 2019. doi: 10.1016/j.cosrev.2019.05.002.

[17] A. Naguib and K. M. Fouad, "Database security: Current challenges and effective protection strategies," in *2024 6th Int. Conf. Comput. Informat. (ICCI)*, Cairo, Egypt, 2024, pp. 120–130.