Journal Publication of International Research for Engineering and Management (JOIREM)
Volume: 04 Issue: 5 | May-2026
ISSN (O) 3107-6696

A Study of Common Network Attacks and Their Prevention Techniques

Kunal [1], Rimmy Chhabra [2], Prince [3], Raghav Aggarwal [4]

[1,3,4] *B.Tech Students, Department of CSE, Quantum University, Roorkee, India.*

[2] *Assistant Professor, Department of CSE, Quantum University, Roorkee, India.*

Abstract - As internet-connected systems grow in their number, network security has become an urgent issue for both organizations and institutions as well as for individuals, thus promoting the need for available and feasible defense options. Although there are several studies which focus on different types of attack (DoS, phishing, SQL injection), there is a lack of a simple, easy-to-learn framework that would cover several attack types from a technical and human-centric perspective. The study of this paper involves literature review of the five common network attacks which are Denial of Service (DoS/DDoS), Man-in-the-Middle (MITM), Phishing, Packet Sniffing and SQL Injection, and also a hands-on experiment of SQL Injection attack using Oracle APEX and its prevention using parameterized queries has been carried out. The result indicates that a three-layered Layered Prevention Framework (LPF) including Awareness Layer, Technical Layer and Monitoring Layer is proposed which will provide a scalable and cost-effective solution to protect the network against the common cyber threats.

Keywords: *Network Security, Cyber Attacks, DoS, MITM, Phishing, SQL Injection, Intrusion Detection, Layered Prevention Framework, Oracle APEX*

1. Introduction

Computer networks are essential for communicating, trading and learning in this globalized society, as well as for government and administration. The internet and digital services have rapidly proliferated, and the threat landscape has grown by orders of magnitude. A consistent finding, as cybercriminals seek to take advantage of vulnerabilities in network systems resulting in data breaches, financial losses, and the disruption of critical services [13].

Reports from different cybersecurity firms indicate that the number of cyber attacks has jumped significantly in the last 10 years. The landscape of threats to organizations is becoming more complex, with attackers coming from all over the world and employing more advanced tools and methods to crack into network security.

In the face of rising concerns about cyberattacks, the majority of networks are vulnerable because they are still operating systems that are outdated, users lack knowledge and awareness, and they don't have cost-effective security methods that are configured for smaller organizations. Most papers that have been published focus on the individual attack type alone, and there is a lack of integrated and practical knowledge for students and novices [6], [13].

1.1 Objectives of the Study

The primary aim of this study is to identify and analyze the five most common types of network attacks, namely DoS/DDoS, Man-in-the-Middle, Phishing, SQL Injection, and Packet Sniffing, with an emphasis on understanding their mechanisms and real-world implications. Alongside this, the study seeks to review and evaluate existing prevention techniques associated with each attack type, assessing their strengths as well as their practical limitations in diverse deployment contexts. A hands-on experiment is also conducted to practically demonstrate SQL injection vulnerabilities using Oracle APEX, followed by the implementation of parameterized queries as a proven countermeasure, providing learners with direct experiential insight into both the attack and its prevention. Building on these theoretical and experimental findings, the study further aims to propose a Layered Prevention Framework (LPF) as a unified, beginner-accessible solution that consolidates human-centric, technical, and monitoring controls into a coherent and scalable defense model suitable for academic institutions and small organizations alike.

1.2 Problem Statement

Despite the availability of various security tools and techniques, network attacks continue to rise due to a lack of user awareness, outdated infrastructure, and the absence of a unified, lightweight prevention framework accessible to non-expert users and small organizations. SQL injection in particular remains one of the most exploited vulnerabilities in database-driven applications, often due to insecure coding practices that are preventable with minimal effort.

2. Literature Review

A number of researchers have investigated network attacks and prevention strategies in various aspects. Mirkovic and Reiher (2004) had some early research that mainly dealt with DoS and DDoS attacks, which led to a taxonomy of attacks and defenses. Later studies by Conti et al. (2016) investigated the possibility of MITM attacks in wireless network and identified vulnerabilities in authentication protocols.

There has been a great deal of research into phishing attacks. Vishwanath et al. (2011) showed that phishing susceptibility is primarily behavioral and thus cannot be addressed by technical measures. Likewise, Clarke (2009) found that input validation is a key defense against SQL injection attacks, but also reported that human error is a significant threat.

Verma and Munjal (2012) have discussed about the packet sniffing and its solutions, highlighting the need of encryption of the data to prevent the data from being intercepted [5], [20]. Personal studies are useful but are usually limited to specific types of attacks. In particular there is a lack of a common framework that integrates awareness-driven and technical solution for various types of attack in a user-friendly model [16], [22]. Moreover, the majority of previous research is only theoretical; very few practical examples of the complex processes of attack and defence have been documented in a controlled setting in the academic environment, thus a lack of experiential knowledge for learners.

3. Methodology

The present study is a blend of theoretical analysis and practical SQL injection experiment to explore vulnerabilities in database driven applications as well as to examine successful preventive mechanism. Why SQL injection was selected for the practical component is intentional: SQL injection is one of the most common and well documented network-layer application attacks and the mechanics of SQL injection can be directly observed and mitigated in a controlled environment. The experiments were performed in Oracle APEX, provided by the institutional Oracle Academy environment, and are easily repeatable in an academic environment without the need for dedicated server infrastructure.

A sample database table has been created to mimic a very basic authentication system, where the username and password fields are used to validate a user's credentials when they log in. SQL queries were run in three different phases, enabling a controlled comparison of the handling of SQL queries in an insecure mode and in a secure mode.

3.1 Experimental Phases

1. Normal Authentication Testing: A standard SQL query was executed using valid credentials to verify the normal login process within the database system, establishing a baseline for comparison.

2. SQL Injection Attack Simulation: A malicious SQL payload was inserted into the password field to manipulate query logic and bypass authentication. This phase directly demonstrates how improper input validation creates exploitable vulnerabilities — the same class of vulnerability responsible for major real-world data breaches.

3. Secure Query Analysis: Parameterized queries using bind variables were implemented and analyzed to demonstrate how secure coding practices prevent SQL injection, connecting the experimental findings directly to the prevention strategies discussed in Section 6.

The outputs generated during each phase were carefully observed and compared. Screenshots of database tables, vulnerable queries, attack simulations, and secure query structures were collected as experimental evidence, forming the empirical basis for the Results and Discussion in Section 7.

4. Common Network Attacks

The following is a description of the five most common impacts of network attacks that have been seen in the modern computing environment. It is included here in the greater taxonomy of attacks, and will be discussed in more detail in the practical experiment of Section 3, because SQL injection is a very important attack.

4.1 Denial of Service (DoS) / Distributed Denial of Service (DDoS)

DoS is an attack in which a server or network is inundated with so much traffic that it is unable to handle it for its intended use by normal users [1]. A DDoS attack is an attack, that is coming from many compromised sources (botnets), where it is more difficult to block. They can cause the downfall of websites, APIs, and online services for long durations.

4.2 Man-in-the-Middle (MITM) Attack

In a MITM attack, the attacker hides and intercepts, and possibly alters the communication between two people that they think are communicating directly [2]. This is typically done over unsecured Wi-Fi networks, employing the ARP spoofing, SSL stripping, and session hijacking techniques.

4.3 Phishing

Phishing is a scam that tricks users into disclosing personally identifying information like passwords, credit cards numbers or log-in information via email, web pages or messages pretending to be from trusted organizations [3, 23]. This takes advantage of human psychology, rather than technical flaws, and so it is not easily defeated by a purely technical defense.

4.4 SQL Injection

SQL Injection is one code injection method that involves malicious SQL statements being inserted into the input fields of an application that interacts with a network [4, 15]. If done, these statements can be used to access, update, or delete data from the back end database without permission. As shown in the experiment in Section 3, even with such a simple payload as ' OR '1'='1' an entire payload can be able to evade password based authentication if the queries are not properly structured.

4.5 Packet Sniffing

Packet Sniffing is a process of capturing information packets while they are moving on a network [5, 20]. Usernames, passwords and other confidential communications, much like conversations, are subject to theft by attackers who rely on specialized software (sniffers) to capture the information from unencrypted communications, particularly on unsecured or shared networks.

4.6 Summary of Attacks

Attack Type	Method	Primary Target	Difficulty Level
DoS / DDoS	Traffic flooding via botnets	Servers & services	Medium – High
Man-in-the-Middle	ARP spoofing, SSL stripping	Communication channels	Medium
Phishing	Fake emails / websites	End users	Low – Medium
SQL Injection	Malicious SQL in input fields	Databases	Medium
Packet Sniffing	Network traffic interception	Unencrypted data	Low – Medium

5. Existing Prevention Techniques

There are many different methods of preventing network attacks to date. Such are briefly summarized below, along with the known limitations, which are directly motivated by the desire for the unified framework proposed in Section 6.

Attack	Existing Prevention Technique	Limitation
DoS / DDoS	Rate limiting, firewalls, CDN-based mitigation	High cost for real-time mitigation at scale [1], [17]
MITM	SSL/TLS encryption, certificate pinning	Complex to implement for small organizations
Phishing	Email filters, anti-phishing browser extensions	Cannot fully counter social engineering
SQL Injection	Input validation, parameterized queries	Requires developer-level expertise; often skipped
Packet Sniffing	VPNs, end-to-end encryption (E2EE)	Performance overhead on low-resource devices

6. Proposed Solution: Layered Prevention Framework (LPF)

The limitations identified in Section 5, combined with the practical insight gained from the SQL injection experiment in Section 3, underscore the need for a prevention approach that is not only technically sound but also human-aware and continuously monitored. This paper proposes a Layered Prevention Framework (LPF) — a three-tier model designed to address multiple network attack types in a unified, cost-effective, and beginner-accessible manner.

Layer 1 — Awareness Layer (Human-Centric)

The first layer is the human weaknesses that technical controls alone can't help, such as phishing attacks and SQL injection weaknesses on the developer's part that were seen in the experiment. It has a number of important components, which are summarized below.

The Awareness Layer is the lowest level of the LPF, designed to tackle the human vulnerabilities which purely technical controls are not sufficient to address. The experiment confirms the significant role human behavior and organizational practices have on the security of an organization, as shown by the SQL injection vulnerabilities discovered on the developer's side but not on the attacker's side, and phishing attacks that were discovered during the experiment. One such goal is to ensure that all network users, from top to bottom, are trained regularly on cybersecurity awareness and are aware of the usual threats and good practices. Periodic phishing simulation drills are performed to educate users to avoid being fooled by phishing emails and links. Developers are adeptly taught to code securely, such as building parameterized queries correctly, as many vulnerabilities are caused by developer-level mistakes. Organizational policies are put in place for password usage and data handling, minimizing variations in handling sensitive information. Also, multi-factor authentication (MFA) is activated as standard practice for all accounts, offering a valuable second-line of defense in the event of compromised credentials [14], [18].

Layer 2 — Technical Layer (Infrastructure-Centric)

The second layer is technical in nature to stop and prevent attacks at the network level and application level. This layer is informed by the experimental findings, particularly the parameterized query as an effective, low cost defence against SQL injection attacks.

The Technical Layer provides the foundation for the LPF with physical controls to prevent and mitigate attacks at the network and application level. This layer is designed to implement the most effective, low cost defenses based on a direct analysis of the experimental results, especially the success of parameterized queries in mitigating SQL-injection attacks, and to implement these defenses without requiring special hardware. Networks are protected at the perimeter with firewalls and Intrusion Prevention Systems (IPS) that examine and reject any malicious traffic attempting to enter critical systems. All data transmitted is protected using SSL/TLS encryption, which means that users and servers can no longer be targeted by Man-in-the-Middle attacks [2], [6]. To protect user connections over insecure or public networks, especially for mobile users and remote workers, Virtual Private Networks (VPNs) are employed. As experimentally shown in Section 3, all web applications must have parameterized queries and extensive input validation to ensure that SQL injection attacks cannot happen at the code level [4], [16], [25]. Finally, rate limiting and traffic filtering technologies are applied at the network edge to absorb and fend off high-volume DoS and DDoS attacks, keeping servers available even against sustained attacks.

Layer 3 — Monitoring Layer (Detection-Centric)

The third layer provides real-time monitoring of network activity, identifying and reacting to threats. The Monitoring Layer is the detection-oriented layer of the LPF that continuously monitors to detect threats that pass through the first two layers of the LPF, and to respond to these threats in real-time. Intrusion Detection Systems (IDS) are used to passively observe all network traffic for abnormalities, known attack signatures and behavioral deviations that are often associated with attacks in progress. Each network interaction is recorded in a single place for post-incident investigations and to provide security teams with a full audit trail in the event of a breach or a violation of the security policy. The automated alerts are set to trigger an alert when administrators notice unusual traffic patterns, unauthorized access attempts, or sudden bursts of data requests, which could indicate a DDoS attack or reconnaissance effort. Vulnerability assessment and penetration testing are also scheduled regularly, to proactively identify and remedy weaknesses found in the network before they are able to be exploited [9], [22]. These monitoring activities collectively make up the LPF a dynamic system that adapts to the rapidly evolving threat landscape, giving organisations the visibility they need to efficiently detect, contain and recover from cyberattacks. (Alheeti and McDonald-Maier, 2018)

6.1 Why LPF is Better than Existing Approaches

Unlike existing solutions that target individual attack types or require enterprise-level resources, the LPF:

- Addresses five common attacks through a single unified framework.
- Combines human awareness with technical and monitoring controls.
- Is scalable and cost-effective for small to medium-sized networks.
- Is grounded in practical experimental evidence, not purely theoretical analysis.
- Requires no specialized hardware — only software tools and training.

7. Results and Discussion

7.1 SQL Injection Experiment Results

The SQL injection experiment was successfully performed using Oracle APEX to analyze authentication vulnerabilities in database-driven applications. The three experimental phases yielded clear, comparable outcomes that validate the need for secure coding practices.

Phase 1: Normal Authentication

A standard authentication query was executed using valid credentials:

Query (Normal Authentication):

```
SELECT * FROM users_demo
WHERE username = 'admin'
AND password = 'admin123'
```

The query returned the authorized user record successfully, confirming that the authentication system functioned correctly under normal conditions. This established the baseline behavior against which the attack scenario was compared.

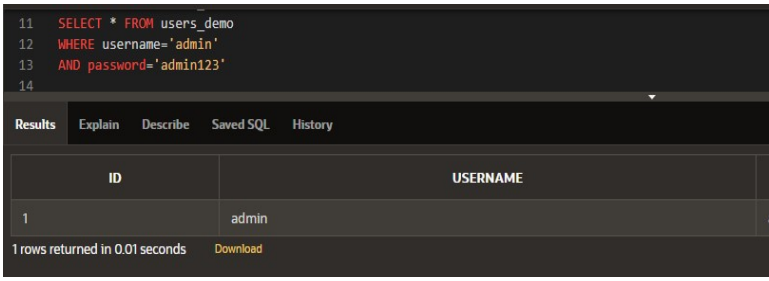

Figure 1

Phase 2: SQL Injection Attack Simulation

A malicious payload was inserted into the password field to manipulate the query logic:

Journal Publication of International Research for Engineering and Management (JOIREM)
Volume: 04 Issue: 5 | May-2026
ISSN (O) 3107-6696

Query (Injection Payload in Password Field):

```
SELECT * FROM users_demo
WHERE username = 'admin'
AND password = '' OR '1'='1'
```

The injected condition ' OR '1'='1' always evaluates to TRUE, causing the database to return all records regardless of whether a valid password was supplied. Authentication was completely bypassed without any correct credentials. This outcome directly demonstrates how a single missing input validation check can expose an entire database to unauthorized access — a vulnerability pattern responsible for high-profile breaches across industries.

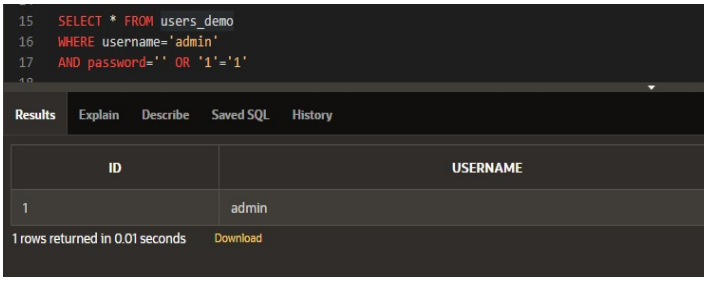

Figure 2

Phase 3: Secure Parameterized Query

To counter the vulnerability observed in Phase 2, a parameterized query using bind variables was implemented:

Query (Parameterized — Secure):

```
SELECT * FROM users_demo
WHERE username = :username
AND password = :password
```

In this approach, user inputs are passed as separate data parameters rather than being embedded directly into the SQL string. The database engine treats the input strictly as data, making it impossible for injected SQL syntax to alter the query's logic. The attack payload that succeeded in Phase 2 was completely neutralized by this approach, with no records returned for invalid credentials.

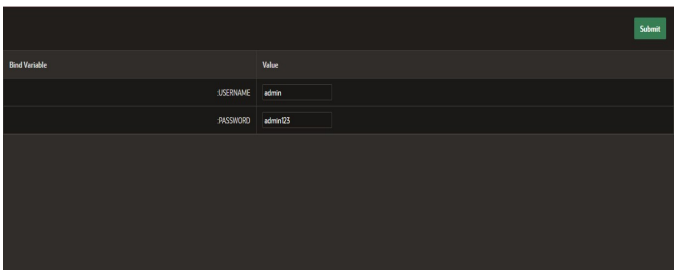

Figure 3

7.2 Discussion

The findings from the experiments clearly demonstrate that SQL injection is not a theoretical threat, but a real one that can be easily done by an attacker with limited skill sets [4], [15], [19]. The three phase experiment was clearly progressive from vulnerable to secure and the defense strategy was more concrete than abstract.

As indicated in the broader study, SQL injection is just one of the types of vulnerabilities that the Layered Prevention Framework (LPF) aims to counter. The Technical Layer sets the parameters for queries as a direct countermeasure. The Awareness Layer solves the root cause: developers who don't know or overlook how to make queries secure. The Monitoring Layer would be able to recognize unusual usage patterns in the database that can be a sign of an injection attack is underway.

The overall analysis of all five attack types echoes this lack of effectiveness of any single prevention mechanism. Phishing attacks are based on human nature and cannot be prevented by a firewall [3], [14]. DDoS attacks are attacks against infrastructure that can't be protected by encryption. This reinforces the LPF structure's multi-layered design - it is a more resilient solution than any single-technique solution.

7.3 Limitations

The results of this study are significant but some limitations are recognized to provide context and give the results their context and application. The SQL injection experiment was performed exclusively in a controlled academic environment on Oracle APEX, and while the results are valid as proof of concept, they do not necessarily represent the complexity that can be faced in different database systems or in a production environment where there is significant variation in configuration, access rights and volume of data. Second, the paper does not attempt to cover APTs or zero day attacks because they require specific countermeasures and real-time threat intelligence that are not within the scope of this study. Third, the success of the Awareness Layer relies on active participation of users, commitment of organizations, and a culture that is security-focused, without which the human-centric aspects of the LPF could have sub-optimal performance. Lastly, the Layered Prevention Framework suggested here has yet to be empirically tested as a whole integrated framework in a live network deployment and is yet to be tested in a wide variety of infrastructure settings in future studies.

8. Comparative Analysis

The following table maps each attack to its existing prevention technique and shows how the proposed LPF improves upon it, incorporating the experimental insights from Section 3 and 7:

Attack	Existing Prevention	LPF Layer	LPF Improvement
DoS/DDoS	Rate limiting, CDN	Technical + Monitoring	IDS alerts + traffic filtering combined
MITM	SSL/TLS	Technical + Awareness	Encryption + user training on public Wi-Fi risks
Phishing	Email filters	Awareness + Monitoring	Phishing drills + MFA enforcement
SQL Injection	Input validation	All 3 Layers	Experimentally validated: parameterized queries (Technical) + developer training (Awareness) + query anomaly alerts (Monitoring)
Packet Sniffing	VPN, E2EE	Technical + Monitoring	VPN policy enforcement + traffic anomaly alerts

9. Conclusion

This paper discussed the five popular network attacks (DoS/DDoS, MITM, Phishing, SQL Injection, Packet Sniffing) with a practical experiment in Oracle APEX that involved SQL Injection. The experiment gave an authentic first-hand look at the insidious threats of insecure query construction, and the collective power of parameterized queries to defuse them.

Based on the literature review and experimental results the Layered Prevention Framework (LPF) was suggested as a consolidated, three-layer framework (Awareness, Technical and Monitoring) which is able to overcome the limitations of the current single-attack, enterprise-focused solutions to address the multi-layered nature of the attacks.Based on the literature review and experimental results, the Layered Prevention Framework (LPF) was proposed as a unified, three-layer framework addressing the multi-layered nature of the attacks, overcoming the limitations of the current single attack, enterprise-focused solutions, which include Awareness, Technical and Monitoring layers. The LPF offers a cost-effective, hands-on and experimentally validated solution, especially for academic institutions, small organisations and for people getting into network security.

Future research would include implementation and empirical evaluation of the whole LPF in a real network environment [17], [21], [22], testing other attacks (MITM, Packet Sniffing, etc.) and using Wireshark in the implementation, and investigating how anomaly detection could be improved with AI in the Monitoring Layer.

References

[1] J. Mirkovic and P. Reiher, "A Taxonomy of DDoS Attack and DDoS Defense Mechanisms," ACM SIGCOMM Computer Communication Review, vol. 34, no. 2, pp. 39–53, 2004.

[2] M. Conti, N. Dragoni, and V. Lesyk, "A Survey of Man in the Middle Attacks," IEEE Communications Surveys & Tutorials, vol. 18, no. 3, pp. 2027–2051, 2016.

[3] A. Vishwanath, T. Herath, R. Chen, J. Wang, and H. R. Rao, "Why Do People Get Phished? Testing Individual Differences in Phishing Vulnerability," Decision Support Systems, vol. 51, no. 3, pp. 576–586, 2011.

[4] J. Clarke, SQL Injection Attacks and Defense, Syngress Publishing, Burlington, MA, 2009.

[5] R. Verma and G. Munjal, "Network Sniffing: Approaches and Defense," International Journal of Advanced Research in Computer Science, vol. 3, no. 4, 2012.

[6] W. Stallings, Cryptography and Network Security: Principles and Practice, 7th ed., Pearson Education, New Jersey, 2017.

[7] K. M. A. Alheeti and K. McDonald-Maier, "Intelligent Intrusion Detection in External Communication Systems for Autonomous Vehicles," Systems Science & Control Engineering, vol. 6, no. 1, pp. 48–56, 2018.

[8] B. Cheswick, S. Bellovin, and A. Rubin, Firewalls and Internet Security: Repelling the Wily Hacker, 2nd ed., Addison-Wesley, Reading, MA, 2003.

[9] OWASP Foundation, "OWASP Top Ten Security Risks," https://owasp.org/www-project-top-ten/, 2021.

[10] Cisco Systems, "Annual Cybersecurity Report," Cisco Press, San Jose, CA, 2023.

[11] Oracle Corporation, "Oracle APEX Documentation," https://apex.oracle.com/en/learn/documentation/, 2024.

[12] N. Provos and P. Honeyman, "Safe Browsing: Detecting and Defeating Web-Based Malware," USENIX Security Symposium, 2004.

[13] R. Anderson, Security Engineering: A Guide to Building Dependable Distributed Systems, 3rd ed., Wiley, Hoboken, NJ, 2020.

[14] M. Donahue and J. Lyle, "Phishing Attacks and Countermeasures," NIST Special Publication 800-177, National Institute of Standards and Technology, Gaithersburg, MD, 2019.

[15] S. Jiang, S. Wu, and Z. Zhang, "Advanced Techniques for Blind SQL Injection Attack Detection," IEEE Transactions on Information Forensics and Security, vol. 16, pp. 1567–1579, 2021.

[16] M. Nasereddin, A. ALKhamaiseh, M. Qasaimeh, and R. Al-Qassas, "A Systematic Review of Detection and Prevention Techniques of SQL Injection Attacks," Information Security Journal: A Global Perspective, vol. 32, no. 4, pp. 252–265, 2023.

[17] P. Kumari and A. Sharma, "A Survey on Machine Learning-Based Intrusion Detection Systems for Network Security," International Journal of Network Security, vol. 22, no. 5, pp. 785–798, 2020.

[18] CISA, "Avoiding Social Engineering and Phishing Attacks," Cybersecurity and Infrastructure Security Agency, https://www.cisa.gov/news-events/news/avoiding-social-engineering-and-phishing-attacks, 2024.

[19] X. Fu and X. Wang, "Out-of-Band SQL Injection: Techniques and Countermeasures," Computers & Security, vol. 105, p. 102262, 2023.

[20] T. Ylonen, "The Secure Shell (SSH) Protocol Architecture," RFC 4251, Internet Engineering Task Force (IETF), 2006.

[21] J. Li and Y. Zhang, "Second-Order SQL Injection Attacks: Detection and Prevention Techniques," Information Sciences, vol. 601, pp. 15–28, 2022.

[22] D. Chou and M. Jiang, "A Survey on Data-Driven Network Intrusion Detection," ACM Computing Surveys, vol. 54, no. 3, pp. 1–36, 2021.

[23] A. Dawabsheh, D. Eleyan, M. Jazzar, and A. Eleyan, "Social Engineering Attacks: A Phishing Case Simulation," International Journal of Scientific & Technology Research, vol. 10, no. 5, 2021.

[24] S. Chowdhury, A. Nandi, M. Ahmad, A. Jain, and M. Pawar, "A Comprehensive Survey for Detection and Prevention of SQL Injection," in Proc. 7th International Conference on Advanced Computing and Communication Systems (ICACCS), vol. 1, pp. 434–437, 2021.

[25] V. Thatikonda and H. R. Mudunuri, "Writing Secure Code in the Digital Age: Preventing Common Vulnerabilities," International Journal of Computer Applications, vol. 185, no. 37, pp. 48–51, 2023.