SECURITY FUNDAMENTALS

CHIJIOKE FRANKLIN EMEJURU

Project Title: Integrated Cybersecurity Strategies for Ensuring Data Integrity, Threat Detection, and Regulatory Compliance

Project Description:
This project explores critical cybersecurity principles and practices essential for safeguarding digital assets and ensuring secure information exchange. It begins with a detailed examination of non-repudiation, emphasizing the use of cryptographic tools to maintain data authenticity and accountability in digital transactions. The study then addresses SQL injection attacks, illustrating how poorly sanitized user input can compromise databases and offering insights into effective prevention strategies. Further, the project evaluates intrusion detection methods, comparing signature-based, anomaly-based, and hybrid monitoring solutions to highlight their respective strengths and weaknesses in identifying and mitigating cyber threats. Finally, it presents an overview of cybersecurity frameworks, with a focus on the PCI DSS standard, showcasing how structured guidelines and best practices help organizations protect sensitive data, meet regulatory requirements, and enhance their overall security posture. Together, these components form a holistic view of modern cybersecurity defense mechanisms.

Non-Repudiation

Non-repudiation is a crucial security principle in information security. The concept demonstrates the source, authenticity, and integrity of data. It guarantees that a party will not invalidate the authenticity of their signature on a document or during the transmission of a message. It acts as evidence that the sender sent the message, and the recipient received it. The principle ensures trust, accountability, and reliability in digital communications and transactions [1]. Understandingly, signatures provide authentication and accountability in traditional paper-based transactions. However, a reliable mechanism for establishing authenticity is more required in the digital realm, where transactions are done remotely and electronically, which points to the importance of non-repudiation.

The principle has several benefits:

1. It helps in establishing accountability. Parties that engage in electronic transactions require assurance that none of them can deny their involvement at a later date. The assurance is facilitated through non-repudiation mechanisms, such as digital signatures and cryptographic techniques, that create a verifiable

transaction record.

2. The principle is crucial for legal and regulatory compliance. Industries such as finance, healthcare, and government departments are bound by stringent regulations concerning the integrity and authenticity of electronic transactions [2]. Non-repudiation mechanisms thus help organizations adhere to the rules

A Scenario Demonstrating Non-Repudiation:

In the case of an online retailer that processes customer orders and payments through its e-commerce platform, non-repudiation serves an important function. In such cases, conserving the integrity and security of the transaction process is essential. For instance, when a customer purchases a product from an online retailer using a credit card, he submits his order and payment through the retailer's website. Subsequently, the transaction details are encrypted and securely transmitted to the retailer's server. When the retailer receives the transaction request, the payment is processed by the server, updates are made on the inventory, and an electronic receipt is produced for the customer. In this case, the customer cannot deny the purchase or file disputes with his credit card company. Moreover, the retailer ensures the authenticity and integrity of the transaction through digital signatures that sign the electronic receipt generated to prove the customer's purchase.

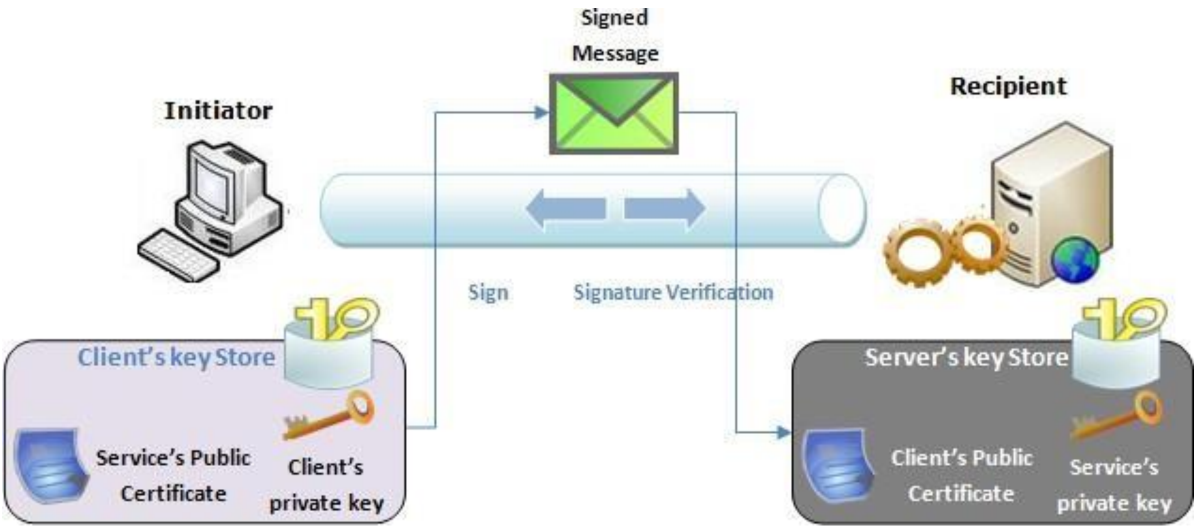


Figure 1: ***(A diagram illustrating non-repudiation)*** **[3].**

SQL:

A SQL (Structured Query Language) Injection attack is a widespread cybersecurity threat that explores the weaknesses in web applications that interact with databases. While conducting such attacks, malicious actors inject malicious SQL code into input fields of web applications [4]. The actors thus trick the application into executing unintended SQL commands. The practices, therefore, enable the actors to have unauthorized access to sensitive data, manipulate database contents and or completely compromise the affected system.

For instance, the attack can be illustrated through a vulnerable e-commerce website that employs SQL queries to access product information from a database. Users can search for products through the website by entering keywords in the search field. However, the website fails to properly sanitize user inputs, exposing them to the vulnerability of SQL injection attacks.

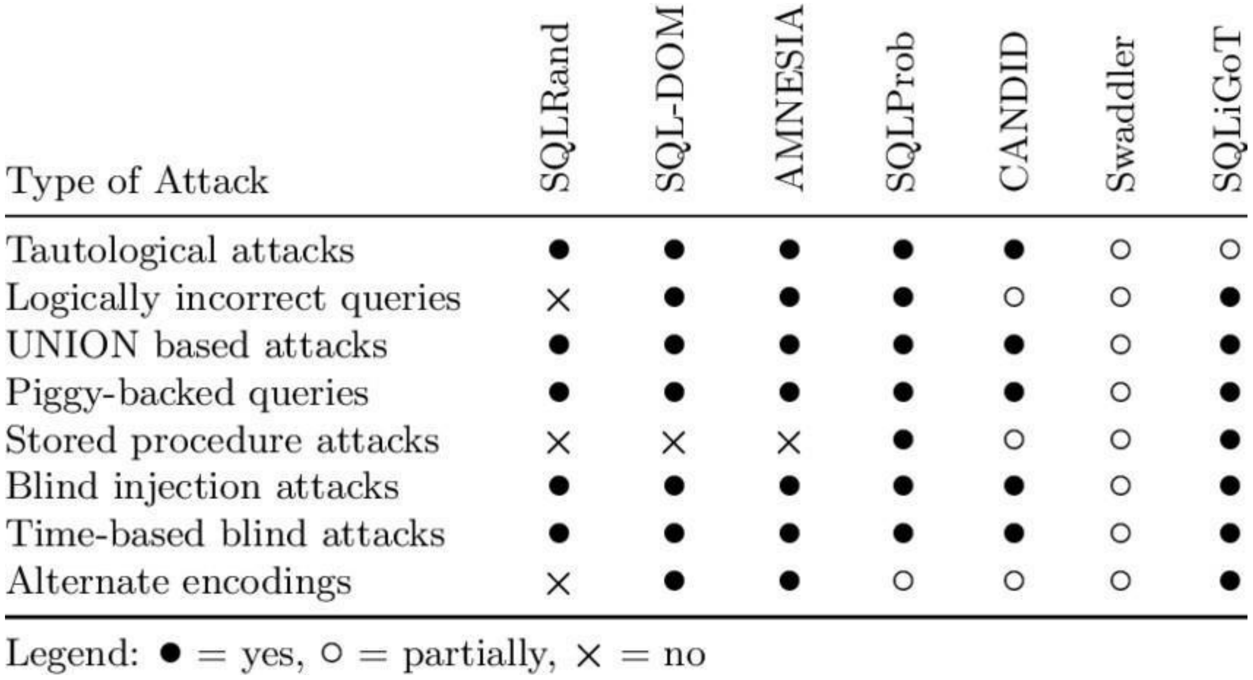
An attacker can enter malicious SQL code into the search field by injecting the vulnerability. For example, the attacker may enter an SQL query that prompts retrieving all records from the database, bypassing any authentication or authorization mechanisms.

```
['; SELECT * FROM Products; --]
```

The attacker's input submission concatenates the web application with the SQL query used to retrieve product information. The result will be an SQL query like the one below.

```
[SELECT * FROM Products WHERE Name LIKE '%'; SELECT * FROM Products; --%']
```

The semicolon (;) ends the original query, and the following query prompts the database to access all records from the 'Products' table. The subsequent two dashes (--) comment out the remaining part of the original query to prevent any syntax errors. When the command is successively processed, the attacker gains access to all products in the database, including sensitive information such as product prices and customer details. If the database has assigned many privileges to the user, it becomes also possible for the attacker to change or delete data on the website. Some attackers may also assume the unauthorized role of controlling the underlying server.



Type of Attack	SQLRand	SQL-DOM	AMNESIA	SQLProb	CANDID	Swaddler	SQLiGoT
Tautological attacks	●	●	●	●	●	○	○
Logically incorrect queries	×	●	●	●	○	○	●
UNION based attacks	●	●	●	●	●	○	●
Piggy-backed queries	●	●	●	●	●	○	●
Stored procedure attacks	×	×	×	●	○	○	●
Blind injection attacks	●	●	●	●	●	○	●
Time-based blind attacks	●	●	●	●	●	○	●
Alternate encodings	×	●	●	○	○	○	●

Legend: ● = yes, ○ = partially, × = no

Table 1: ***(A Table Showing Types of SQL Attacks)*** **[4].**

Signature Information-based, Anomaly-based, and Hybrid- based monitoring solutions and their Strengths and Weaknesses:

Intrusion Detection/Prevention Systems (IDS/IPS) and fundamental elements of cybersecurity infrastructure. These components are involved in identifying and mitigating threats to network security. The three commonly employed solutions within the framework are signature Information-based, Anomaly-based, and Hybrid-based approaches. Each approach has its advantages and limitations.

Signature Information-Based Monitoring

Signature Information-based monitoring depends on predefined patterns or signatures of known attacks to detect and prevent intrusion. The signatures are vital digital fingerprints representing malicious activities such as virus patterns and network packet structures. These activities lead to common attacks such as denial-of-service (DoS) or SQL injection [5]. The approach is commended for accurately detecting known threats with minimal false positives. The strength is backed by its signatures, based on well-understood attack techniques. However, the approach's main drawback is its reliance on initial knowledge of threats. It cannot be relied on in novel or zero-day attacks without signatures.

Anomaly-Based Monitoring

Anomaly-based monitoring entails the identification of deviations from normal network behaviour instead of focusing on attack signatures. With this approach, establishing a standard of normal network behaviour helps spot any changes from the starting point as possible attacks. The main advantage of these systems is their ability to detect past unknown threats. Consequently, they are preferred in environments where zero-day attacks are a significant problem. However, they have a limitation of being more prone to false positives. The issue is driven by legitimate activities that shift from established patterns to trigger alerts.

Hybrid-Based Monitoring Solutions

Hybrid-based monitoring solutions combine signature and anomaly-based approaches' strengths while reducing their weaknesses. The solutions leverage a hybrid approach to achieve higher accuracy and improve the detection of threats [6]. The hybrid approach is advantageous due to its more comprehensive threat coverage and reduced likelihood of false positives. However, the approach has a limitation related to the need for careful integration of disparate detection mechanisms.

Cybersecurity Framework:

A cybersecurity framework is a structured set of guidelines, best practices, and standards developed to aid organizations in establishing, implementing, and maintaining effective cybersecurity measures [7]. By adopting a cybersecurity framework, an organization can enhance its cybersecurity posture and minimize the likelihood of security and data breaches. It also ensures that organizations comply with regulatory requirements and improves their trust among customers and stakeholders.

Benefits Of Adopting a Cybersecurity Framework:

There are broad benefits to organizations' adoption of a cybersecurity framework. For instance, the framework offers a structured approach to cybersecurity management. It provides organizations a roadmap for identifying, assessing, and prioritizing cybersecurity risks. When organizations comply with cybersecurity regulations such as the Payment Card Industry Data Security Standard (PCI DSS) and General Data Protection Regulation (GDPR), they can avoid penalties, lawsuits and reputational damage.

The PCI DSS, its Purpose and Objective

The PCI DSS framework is a set of security standards developed to maintain a secure environment during processing, storing, or transmitting credit card information. The main purpose of the framework is to safeguard cardholder data from theft and fraud [8]. The framework aims to improve payment card data security by combining technical controls, policies, and procedures.

The Components of the PCI DSS

The PCI DSS has various components, such as creating and maintaining a secure network, safeguarding cardholder data, and implementing a vulnerability management program. Building and maintaining a secure network involves strong security mechanisms such as firewalls. The firewalls aim to protect cardholder data from unauthorised access and malicious activity. The protecting cardholder data component entails the implementation of encryption and other security measures for safeguarding sensitive cardholder data both in transit and at rest. The program for maintaining vulnerability management underscores the need to regularly identify, assess, and address security vulnerabilities.

Summary:

Non-repudiation is a key principle in cybersecurity that ensures a party cannot deny the authenticity of their actions, such as sending a message or approving a transaction, typically enforced through digital signatures and cryptographic tools. It promotes trust, accountability, and legal compliance in digital environments. SQL injection is a common cyberattack where malicious SQL code is inserted into input fields to manipulate databases, potentially exposing or altering sensitive information, highlighting the importance of secure coding and input validation. Intrusion monitoring techniques include signature-based systems that detect known threats, anomaly-based systems that identify deviations from normal behavior, and hybrid systems that combine both for improved accuracy and coverage, though each has trade-offs in terms of false positives and adaptability. Cybersecurity frameworks, like PCI DSS, provide structured guidelines and standards to help organizations protect sensitive data, comply with regulations, and maintain strong security postures by implementing secure networks, encrypting data, and managing vulnerabilities.

References

[1] W. Fang, W. Chen, W. Zhang, J. Pei, W. Gao, and G. Wang, “Digital signature scheme for information non-repudiation in blockchain: a state-of-the-art review,” *EURASIP Journal on Wireless Communications and Networking*, vol. 2020, no. 1, Mar. 2020, doi: 10.1186/s13638-020-01665-w.

[2] K. Banerjee and S. Saha, “International Journal of Computing and Digital Systems,” *International Journal of Computing and Digital System/International Journal of Computing and Digital Systems*, Feb. 2019, doi: 10.12785/ijcds.

[3] F. P. Estrada and R. G. Lázaro, “WSO2 developer’s guide,” O’Reilly Online Learning,, Accessed: May 3, 2024 . [Online]. Available: https://www.oreilly.com/library/view/wso2-developers-guide/9781787288317/70805ef5-e2ae-4698-90f6-780df9c98d2c.xhtml.

[4] D. Kar, S. Panigrahi, and S. Sundararajan, “SQLIGOT: Detecting SQL injection attacks using graph of tokens and SVM,” *Computers &amp; Security*, vol. 60, pp. 206–225, Jul. 2016. doi:10.1016/j.cose.2016.04.005.

[5] A. Binbusayyis, “Hybrid VGG19 and 2D-CNN for intrusion detection in the FOG-cloud environment,” *Expert Systems With Applications*, vol. 238, p. 121758, Mar. 2024, doi: 10.1016/j.eswa.2023.121758.

[6] Y. Jia, M. Wang, and Y. Wang, “Network intrusion detection algorithm based on deep neural network,” *IET Information Security*, vol. 13, no. 1, pp. 48–53, Jan. 2019, doi: 10.1049/iet-ifs.2018.5258.

[7] A. Kumar, P. Pranav, and P. Pranav, “Analysis of SQL injection attacks in the cloud and in WEB applications,” *Security and Privacy*, vol. 7, no. 3, Jan. 2024, doi: 10.1002/spy2.370.

[8] S. Lincke, *Information Security planning*. 2024. doi: 10.1007/978-3-031-43118-0.