Database Security-Inference and Flow Control

Team - Amulya

Sai Jonnalagadda, Julie Jackson

and Prashant Tomar

Agenda

- Introduction
- Importance of Database Security
- Fundamentals of Inference Control
- Fundamentals of Flow Control
- Threats and Vulnerabilities
- Inference Control Techniques
- Flow Control Mechanisms
- Case Studies
- Best Practices
- Conclusion

Introduction

- Database security: Guarding data against unauthorized access, alteration, or destruction.
- Inference Control: Preventing unauthorized insights from data analysis.
- Flow Control: Managing data dissemination to uphold security policies.

Importance of Database Security

- Data breaches can result in significant financial losses and erode customer trust.
- Regulatory compliance: A necessity for legal and privacy standards adherence.
- Integral to maintaining data integrity, confidentiality, and availability.

Fundamentals of Inference Control

Prevents unauthorized deduction of sensitive information from available data.

Direct Inference: Collecting confidential details without violation of access controls.

Indirect Inference: Assembling data points to uncover sensitive information.

Fundamentals of Flow Control

REGULATES DATA TRANSFER BETWEEN VARYING LEVELS OF SECURITY DOMAINS.

ESSENTIAL FOR ENFORCING INFORMATION CONFIDENTIALITY AND INTEGRITY POLICIES.

FOUNDATIONAL TO MULTI-LEVEL SECURITY MODELS, E.G., BELL-LAPADULA.

Threats and Vulnerabilities

1

Common threats include SQL injection, privilege escalation, and insider misuse.

2

Vulnerabilities arise from weak authentication, improper configuration, and inadequate access controls.

3

Inference attacks exploit the aggregation or correlation of data to reveal secrets.

Inference Control Techniques

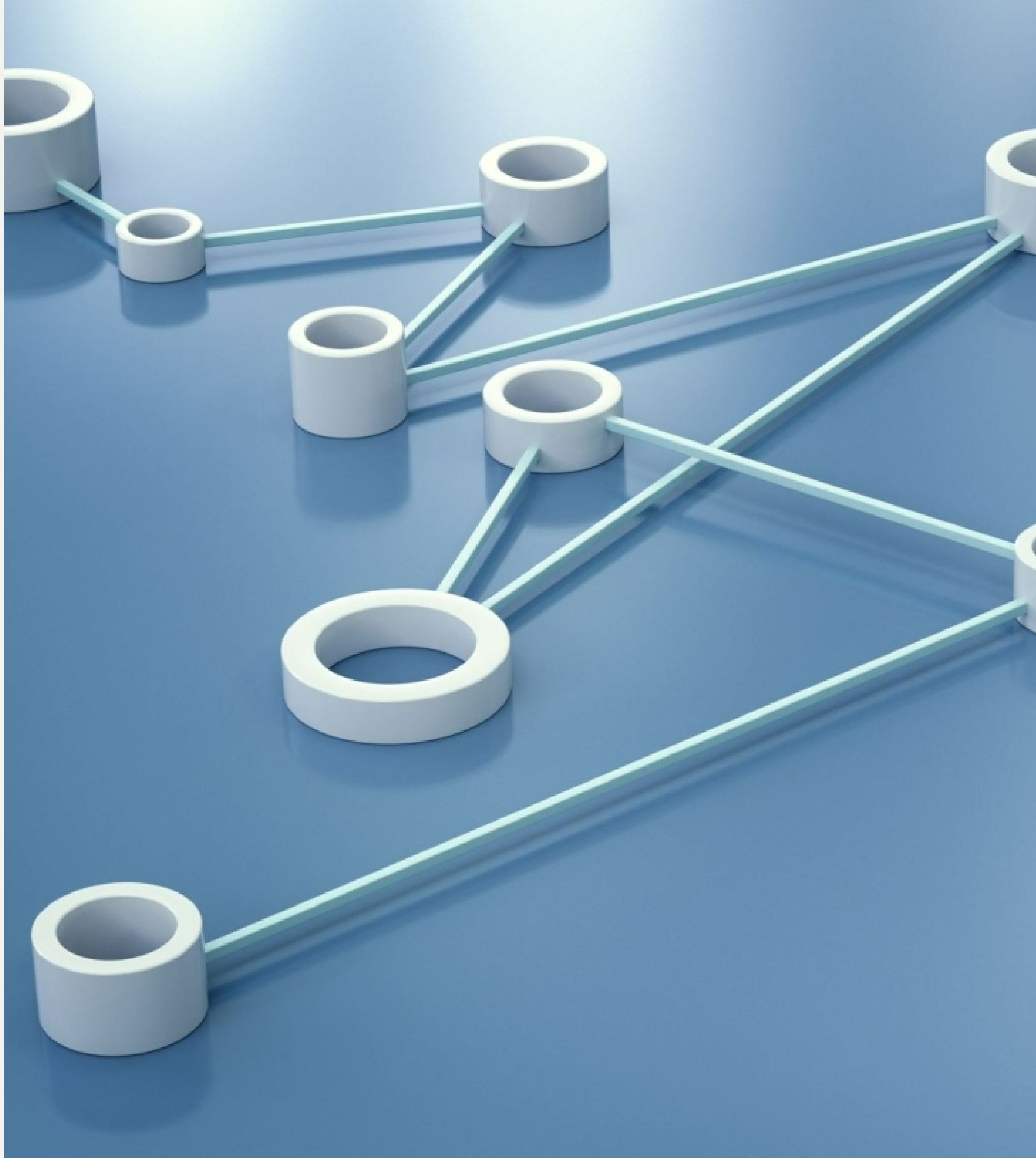
Noise and Perturbation: Adding randomness to data to obscure sensitive details.

Query Restrictions: Limiting the nature and scope of database queries to prevent inference.

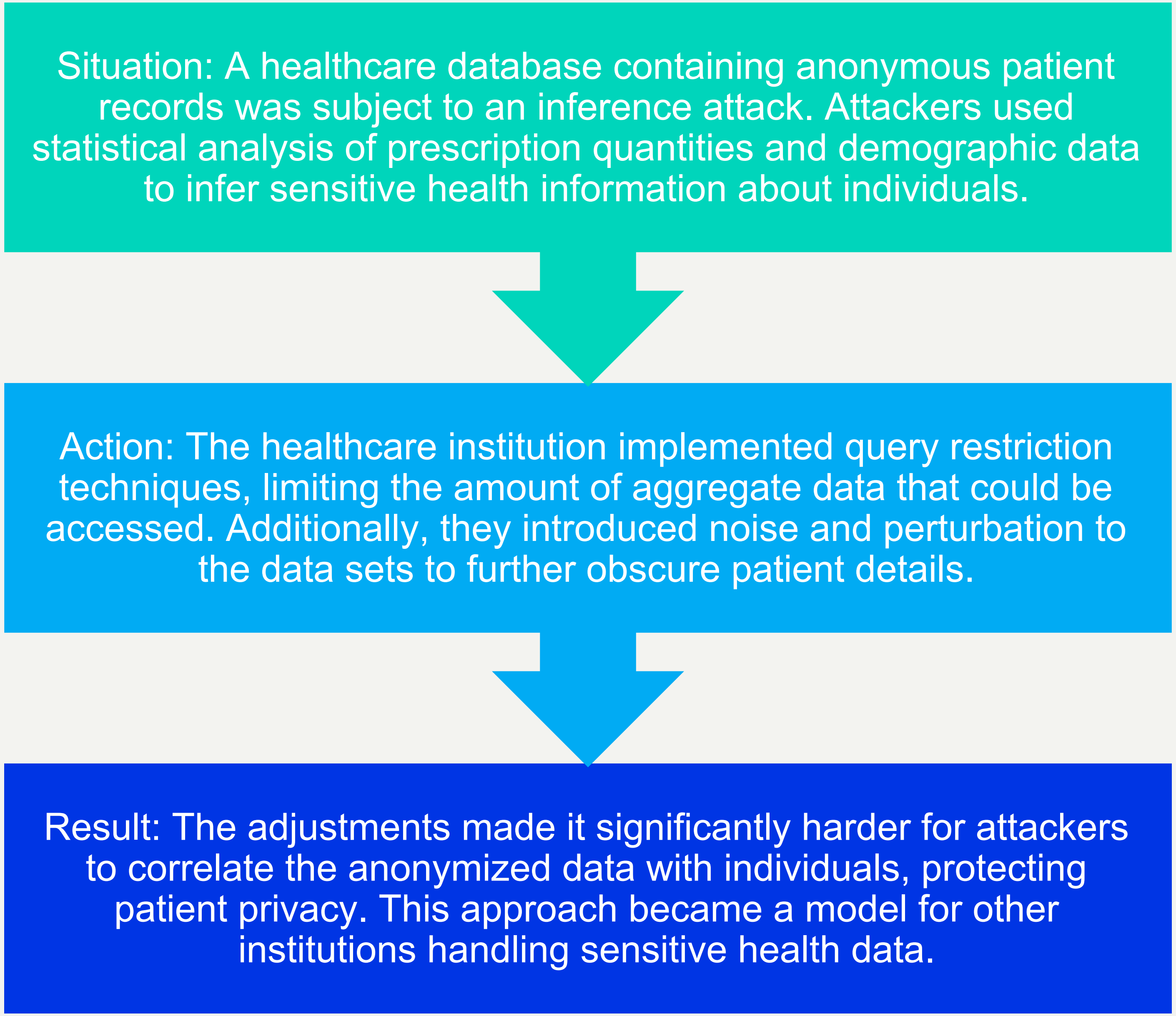
Polyinstantiation: Creating multiple records with the same primary keys but different data depending on clearance.

Flow Control Mechanisms

- Security Labels: Assigning classifications to data and users to dictate access permissions.
- Bell-LaPadula Model: Prevents data from flowing from higher to lower security levels.
- Biba Model: Ensures information flow does not degrade data integrity.



Case Study: Inference Control

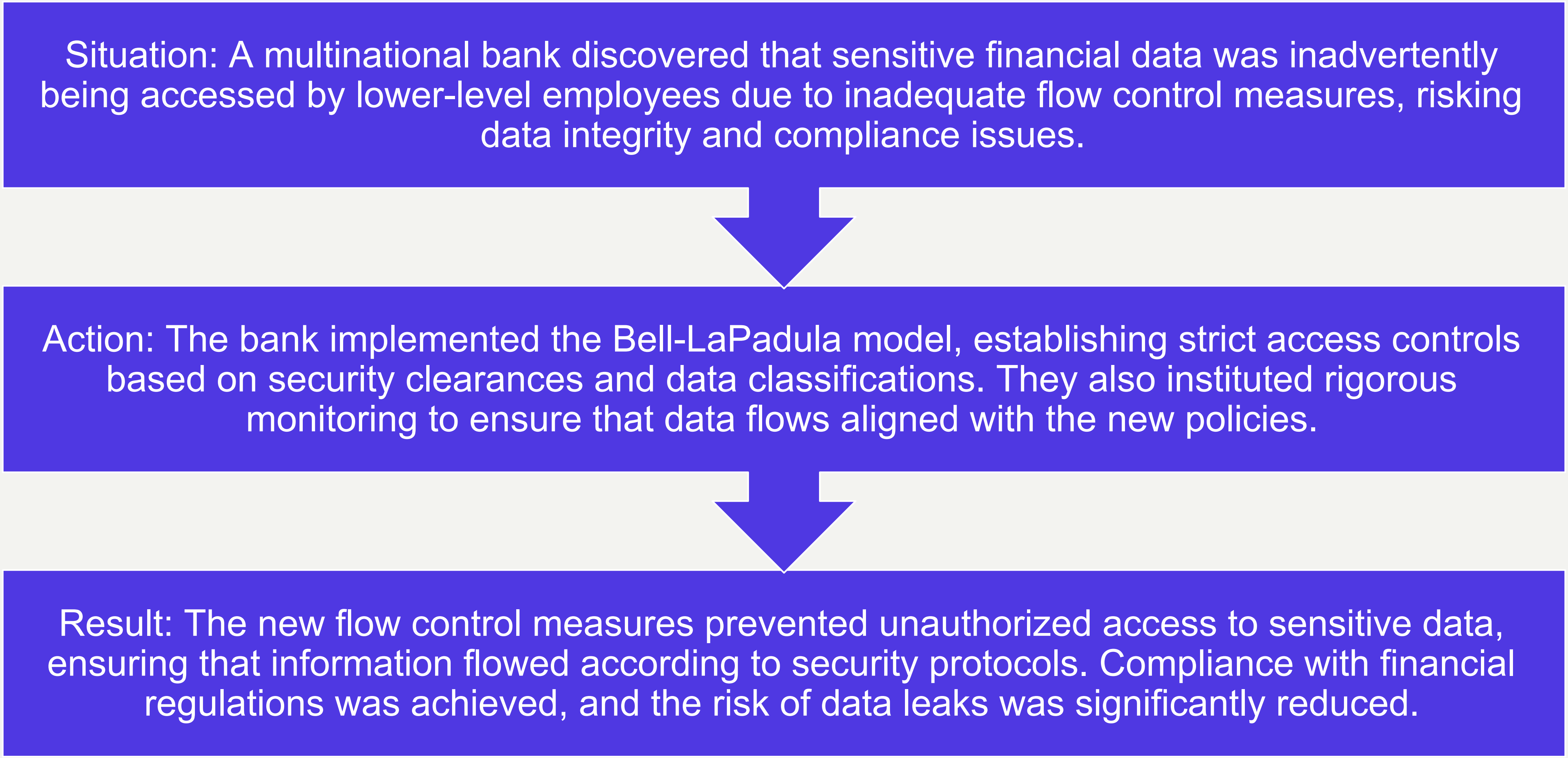


Situation: A healthcare database containing anonymous patient records was subject to an inference attack. Attackers used statistical analysis of prescription quantities and demographic data to infer sensitive health information about individuals.

Action: The healthcare institution implemented query restriction techniques, limiting the amount of aggregate data that could be accessed. Additionally, they introduced noise and perturbation to the data sets to further obscure patient details.

Result: The adjustments made it significantly harder for attackers to correlate the anonymized data with individuals, protecting patient privacy. This approach became a model for other institutions handling sensitive health data.

Case Study: Flow Control



Situation: A multinational bank discovered that sensitive financial data was inadvertently being accessed by lower-level employees due to inadequate flow control measures, risking data integrity and compliance issues.

Action: The bank implemented the Bell-LaPadula model, establishing strict access controls based on security clearances and data classifications. They also instituted rigorous monitoring to ensure that data flows aligned with the new policies.

Result: The new flow control measures prevented unauthorized access to sensitive data, ensuring that information flowed according to security protocols. Compliance with financial regulations was achieved, and the risk of data leaks was significantly reduced.

Regular Audits and Monitoring

Schedule frequent security audits to identify vulnerabilities.

Monitor database activity to detect unusual access patterns that may indicate inference attacks.

Use audit trails to investigate and remediate security incidents promptly.

Strong Access Control Policies

Implement least privilege access to minimize unnecessary data exposure.

Enforce authentication and authorization to control user activities and access levels.

Update access rights dynamically with changes in user roles or responsibilities.

Encryption of Data at Rest and in Transit

Use strong encryption standards to protect data on disk and during transmission.

Ensure encryption keys are managed securely and rotated regularly.

Apply encryption selectively based on data sensitivity to optimize performance without compromising security.

Secure Database Design and Development

Integrate security into the database design phase—'security by design'.

Avoid database designs that can be easily exploited through inference attacks.

Regularly update and patch database management systems to close security gaps.

Security Training for Staff

Conduct regular security awareness training for all staff members.

Specialized training for IT staff on recognizing and preventing inference attacks.

Promote a culture of security mindfulness to protect against social engineering threats.

Use of Database Activity Monitoring Tools

Deploy tools that provide real-time monitoring of database transactions.

Set alerts for suspicious activities that could indicate potential inference or unauthorized data flows.

Automate response mechanisms for rapid containment of security incidents.

Implementation of Data Masking Techniques

Mask data to obscure sensitive information from unauthorized users.

Apply data masking dynamically for users with different access levels.

Ensure that masking techniques do not introduce inference vulnerabilities.

Applying Comprehensive Data Governance

1. Establish a data governance framework to manage data throughout its lifecycle.

2. Define clear policies for data classification, handling, and retention.

3. Ensure policies are enforceable and compliant with relevant laws and regulations.

Conducting Impact Assessments and Testing

01

Perform regular impact assessments to understand the potential damage of inference and flow control breaches.

02

Test security measures through penetration testing and red team exercises.

03

Use the results to refine and strengthen database security postures.

Conclusion

Recap: Database security is critical for protecting sensitive information from unauthorized access and ensuring compliance with data protection regulations.

Key Takeaways: Effective inference control and flow control measures are essential to mitigate risks of data leaks and unauthorized data deductions.

Call to Action: Implement robust security policies, regular audits, and continuous monitoring to safeguard your database against emerging threats and vulnerabilities.