Incident Response Playbook: Web Application SQL Injection (SQLi)

Team AnubisX

Version 1.0
September 23, 2025

Document Control

Attribute	Value
Version	1.0
Status	Draft
Owner	AnubisX Security Team
Review Cycle	Every Quarter

Contents

1 Introduction

1.1 Purpose

This playbook defines incident response procedures for handling ”Web Application SQL Injection (SQLi)”. It provides roles, responsibilities, detection indicators, containment steps, and recovery guidance to minimize impact and restore services.

1.2 Scope

This playbook applies to systems, network components, cloud services, and personnel. It is intended for use by incident responders, SOC analysts, IT operations, legal, and leadership.

2 Overview of the Attack

SQL Injection targets web applications by injecting malicious SQL to read/modify database contents or escalate privileges. Key risks include:

- Exfiltration of sensitive data
- Database integrity compromise
- Remote code execution in some app stacks

3 Incident Response Phases

This playbook follows the NIST Incident Response lifecycle framework.

3.1 Phase 1: Preparation

Goal: To ensure the team is equipped and ready to respond to a SQL injection incident before it occurs.

- **Roles and Responsibilities:** Define roles: Incident Commander, Lead Analyst, Forensics, IT, Communications.
- **Logging & Auditing:** Ensure logging and centralized authentication audits are enabled.
- **Tools Resources:** Deploy specialized detection rules and maintain playbooks for the specific alert type.
- **Training:** Regular backups and least-privilege access models.

3.2 Phase 2: Identification & Analysis

Goal: Confirm the activity and determine scope and severity.

1. **Initial Analysis and IOC Evaluation:** Analyze logs and alerts to identify Indicators of Compromise (IOCs). Common IOCs include:
    - Unusual SQL errors in web logs
    - Queries with tautologies or UNION SELECT patterns
    - Unexpected large result sets or delay-based responses

2. **Severity Level Assessment:** Classify the incident to ensure appropriate allocation of resources. Severity is based on: Operational Impact, Criticality of affected systems/data, Scope of attack, and Detection/Recovery timelines (MTTD/MTTR).

Level	**Description**	**Example**	**MTTD**	**MTTR**
Low	Single endpoint shows suspicious query patterns.	A dev/test endpoint has accidental misconfig causing SQL errors.	<4 hrs	<24 hrs
Medium	Multiple endpoints exploited with limited data exposure.	SQLi used to dump small tables but not admin data.	4-12 hrs	1-3 days
High	Significant data exfiltration or admin user exposure.	Large tables containing PII accessible via injection.	12-48 hrs	3-7 days
Critical	Complete DB compromise and remote code execution leading to server takeover.	Attack leads to full DB admin access and web shell deployment.	48+ hrs	7-21 days

Table 1: Incident Severity Matrix

3.3 Phase 3: Containment

Goal: To limit attacker actions and preserve evidence.

- WAF rules to block payloads, temporarily disable vulnerable endpoints.
- Take DB snapshots, restrict DB access, rotate credentials.

3.4 Phase 4: Eradication

Goal: To remove malicious components and prevent reinfection.

- Patch web application code, parameterize queries, apply principle of least privilege to DB accounts.

3.5 Phase 5: Recovery

Goal: To safely restore systems and business operations.

- Restore DB from clean backups if integrity compromised.
- Validate no web shells or backdoors remain.

3.6 Phase 6: Post-Incident Activities (Lessons Learned)

Goal: To strengthen resilience and prevent recurrence.

- Conduct a blameless post-mortem and update playbooks.
- Produce final incident report and recommended mitigations.
- Implement controls to reduce recurrence.

4 MITRE ATT&CK Framework Mapping

SQL Injection ATT&CK Mapping

- **Tactic: Initial Access**
  - *T1190 – Exploit Public-Facing Application*
  - *T1078 – Valid Accounts*
- **Tactic: Impact**
  - *T1537 – Transfer Data to Cloud Account*
  - *T1486 – Data Encrypted for Impact*