**Detecting SQL Injection in Source Code:**

1. **Static Code Analysis Tools:** Use static analysis tools like Fortify, Checkmarx, or SonarQube that can scan your source code for potential SQL Injection vulnerabilities. These tools analyze code without executing it and can identify suspicious patterns or code constructs that may lead to SQL Injection.
2. **Manual Code Review:** Conduct a manual review of your codebase to search for potential SQL Injection points. Look for instances where user input is concatenated directly into SQL queries without proper validation and sanitization.
3. **Security Code Review Checklist:** Follow a security code review checklist that includes SQL Injection detection techniques. Common signs include the use of unsanitized user input in SQL queries, lack of prepared statements, and dynamic query construction based on user input.

**Detecting SQL Injection in Running Applications:**

1. **Web Application Firewalls (WAFs):** Implement a WAF as part of your application's security infrastructure. WAFs can detect and block SQL Injection attempts in real-time by analyzing incoming HTTP requests for malicious patterns and known attack vectors.
2. **Intrusion Detection and Prevention Systems (IDPS):** Use IDPS to monitor network traffic and detect patterns indicative of SQL Injection attacks. These systems can raise alerts or take preventive actions when suspicious activity is detected.
3. **Database Auditing:** Enable database auditing features if available (e.g., MySQL Enterprise Audit, Oracle Database Auditing). These features can log all SQL queries executed against the database, allowing you to review and detect any unusual or unauthorized queries.
4. **Logging and Monitoring:** Implement comprehensive logging and monitoring in your application. Monitor application logs for unusual SQL query patterns or errors that might be indicative of SQL Injection attempts.
5. **Behavioral Analysis:** Employ behavioral analysis tools that can learn the normal behavior of your application and detect anomalies. Sudden spikes in database activity or unusual query patterns can be indicative of SQL Injection attacks.
6. **Input Validation and Output Encoding:** Implement input validation and output encoding mechanisms in your application to prevent SQL Injection in the first place. If an input contains unexpected characters or patterns, you can log or block the request.