

# SQL Injection Incident Response Playbook

Title	SQL Injection Playbook
Version	V1
Date issued	DD-MM-YYYY
Status	In progress
Document owner	Full Name
Creator name	Full Name
Creator organization name	<Organization Name>
Subject category	SQL Injection Incident Response
Access constraints	NA
Review cycle	Annually

## 1. Introduction

### 1.1 Incident Overview

Attackers execute malicious SQL queries or SQL statements from web applications to compromise the target database. Attackers can successfully launch this attack when an application does not appropriately validate the input before passing an SQL statement. They can execute SQL injection attacks from the web browser's address bar, form fields, queries, and web searches. Upon successful execution, attackers can perform malicious tasks such as manipulating database contents, running malicious codes against the database, and logging into applications without valid credentials.

Assume that CyberZone Solutions received a security alert from one of its security solutions indicating a malicious input that contained an SQL query aimed at accessing the database server. The alert was reported to the service desk for generating a ticket and they escalated the incident to the concerned IH&R team.

### 1.2 Purpose of Playbook

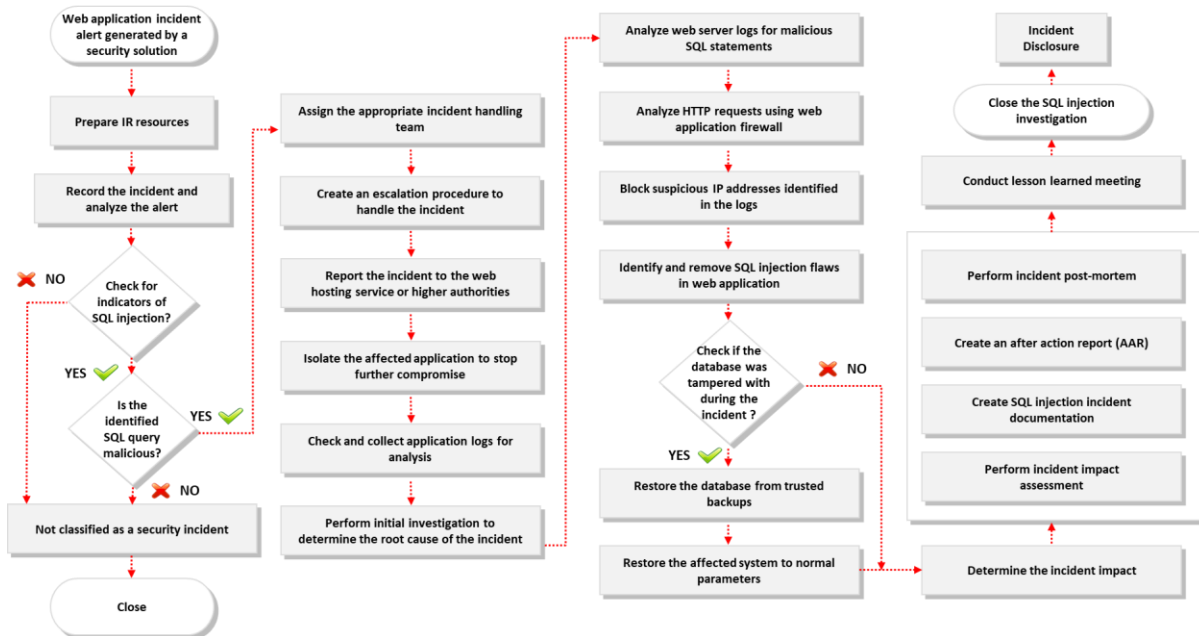
The main purpose of this playbook is to provide guidance on executing the incident response process while handling SQL injection attacks. This playbook includes step-wise guidance for the IH&R team to implement mitigative actions and defend the organizational network against SQL injection attacks.

### 1.3 Scope

This playbook is developed for the use of incident responders to handle and respond to SQL injection incidents in an organization. Additionally, this document must be used alongside the incident response plan of the organization. The scope of this document is listed below (not limited to):

- Determine the business impact caused by the SQL injection incident
- Determine the traffic path through which SQL injection occurred
- Determine the source IP address of the attack traffic
- Understand and document various vulnerabilities that resulted in an SQL injection incident
- Identify any related activities by checking the following:
  - Alerts and notifications from tools such as WAF, SIEM, and IDS
  - Determine possible payload injected for compromising the database
  - Anomalies in log files such as web server, application, and database logs
  - Abnormal behavior in web applications such as unexpected pop-ups and spam messages
  - Changes in the passwords of existing accounts
  - Increase in form submission errors
  - Leakage of sensitive data
- Implement the incident response plan under the supervision of higher authorities of the organization
- Detect and analyze the SQL injection incident
- Implement the remediation steps in an efficient and timely manner

## 1.4 Workflow Diagram



Workflow diagram for SQL injection incident response

## 2. Preparation

### 2.1 Objectives

The main objectives of the preparation phase are listed below:

- Prepare the organization to respond to SQL incidents in a timely and effective manner
- Define the roles of various personnel and their communication medium for the entire SQL injection incident response process
- Prepare organizational systems, network, and data for SQL injection incidents
- Prepare employees regarding their roles and reporting procedures during an SQL injection incident

### 2.2 Activities Involved

*[Activities may differ based on organizational policies, but they are not limited to the following.]*

- Prepare for SQL injection incident response:
  - Keep the IH&R team updated about the latest web application attacks and threats by following relevant journals, magazines, web resources, CSIRTs, and other organizations
  - Develop and maintain a backup website to publish the organization's content
  - Maintain the continuity of internet services with the help of ISPs

- Implement out-of-band communications because regular communication resources can be compromised by the attacker
- Provide easy access to the required documentation such as incident response plan and network architecture for responding to an SQL injection incident. Links of important document are given below:
  - Reference 1:
  - Reference 2:
  - Reference 3:
- Ensure that the IH&R team can export the web server's log files to an external server for review and analysis when required
- Ensure that the IH&R team is aware of applications that belong to the organization, stored data, accounts associated with these data, people with access to these data, and type of data security implemented
- Implement additional security features in websites and web applications to protect sensitive information from being compromised
- Configure web application firewalls (WAFs) such as dotDefender, AppTrana WAF, and FortiWeb, to protect web applications from SQL injection incidents
- Configure security incident and event management (SIEM) solutions such as ArcSight Enterprise Security Manager (ESM) and AlienVault OSSIM to efficiently log, analyze, and alert security incidents
- Use web application security incident detection tools such as Alert Logic MDR and IllusionBLACK to detect SQL injection incidents in advance
- Configure web application and network monitoring tools to alert the IH&R team in case of suspicious events
- Configure access controls for website login pages to protect sensitive information
- Create and maintain a disaster recovery plan
- Create a whitelist of IP addresses and protocols to perform the incident response process
- Maintain an inventory of organizational IT Infrastructure to determine active assets and their whereabouts
- Inform the employees:
  - Create a proper format and mechanism for reporting and registering similar complaints
  - Ensure training and awareness sessions for developers regarding proper coding practices to mitigate SQL injection attacks

- Provide proper contact information of personnel who can be contacted by employees in case of an SQL injection incident, along with their communication channels

### 2.3 Stakeholders Involved/Communication Established

The stakeholders involved in the aforementioned activities and their communication modes are listed below:

Activities	Stakeholders Involved	Communication Mode/Channel
Prepare for SQL injection incident response <ul style="list-style-type: none"> <li>○ Create incident response processes and procedures</li> <li>○ Define roles and responsibilities</li> <li>○ Review recent incident reports</li> <li>○ Incorporate threat intelligence</li> <li>○ Maintain network architecture and data flow diagrams</li> <li>○ Define threat indicators and incorporate alerting solutions</li> </ul>	CISO	Email, Phone, Text Message
	Information Security Manager	Email, Phone, Text Message
	IT Manager/Director	Email, Phone, Text Message
	Service Desk	Email, Phone, Text Message
	Service Delivery Manager	Email, Phone, Text Message
	IH&R Team	Email, Phone, Text Message
	Administrators	Email, Phone, Text Message
	Legal Team	Email, Phone, Text Message
	Federal Agency	Email, Phone, Text Message
	Business Continuity Lead	Email, Phone, Text Message
Inform the employees	Information Security Manager	Email, Phone, Text Message
	IT Manager/Director	Email, Phone, Text Message
	HR Manager/Director	Email, Phone, Text Message
	Administrators	Email, Phone, Text Message
	IH&R Team	Email, Phone, Text Message

## 2.4 Additional Information (if any)

**Note:** Refer to the following documents to fill the necessary details:

- a. Preparation to Web Application Security Incidents Checklist.docx
- b. IH&R Plan Template.docx

## 3. Detection and Notification

### 3.1 Objectives

The main objective of this phase is to identify SQL injection incidents and report it to the appropriate incident handling team for further analysis.

### 3.2 Activities Involved

*[Activities may differ based on organizational policies, but they are not limited to the following.]*

- Detect the SQL injection incident through initial investigation:
  - Check for alerts originating from web application security incident detection tools such as Alert Logic MDR and IllusionBLACK
  - Check for alerts and notifications from tools such as WAF, SIEM, and IDS
  - Check for possible SQL injection payloads in web application traffic
  - Check for anomalies in log files such as web server, application, and database logs
  - Check for unusual tables within the database
  - Check for unusual entries within the tables
  - Check for unusual permissions or ownership changes
  - Check for unusual login/logouts times
  - Check database logs for multiple errors within in a short duration
  - Check for the following error codes:
    - Errors 102 and 105: Unusual syntaxes in query
    - Error 205: Union command abuse to determine the number of columns in a table
    - Error 245: Attempts to retrieve the target database's name
    - Errors 208 and 2812: Access attempts to stored procedures or objects that are invalid
    - Error 18456: Failed login attempts
  - Check for extended events such as **sqlserver.error\_reported**

- Run the default event session such as **system\_health** to detect advanced errors
- Check HTTP request traffic for abnormal activities containing SQL queries
- Use URL decoders to determine the type of SQL injection attack
- Check whether sensitive information such as user passwords and file servers has been leaked
- Check for suspicious activities in user accounts such as new processes, users, and jobs
- Check if the server is receiving too many requests for accessing the same file
- Check for the creation of new admin-level or FTP accounts
- Check for unfamiliar error messages such as deprecated functions and undefined offsets
- Notify other organizations about the incident and involve legal authorities in case of data breach

### 3.3 Stakeholders Involved/Communication Established

The stakeholders involved in the aforementioned activities and their communication modes are listed below:

Activities	Stakeholders Involved	Communication Mode/Channel
Detecting the incident <ul style="list-style-type: none"> <li>○ Monitor security solutions</li> <li>○ Respond to manual and automated alerts</li> <li>○ Escalate the incident via the ticketing system (if not escalated)</li> </ul>	Information Security Manager	Email, Phone, Text Message
	IH&R Team	Email, Phone, Text Message
	Resilience Lead	Email, Phone, Text Message
	Business Continuity Lead	Email, Phone, Text Message
	Policy Area Lead	Email, Phone, Text Message
Initial investigation <ul style="list-style-type: none"> <li>○ Collect initial evidence data</li> </ul>	Information Security Manager	Email, Phone, Text Message
	IH&R Team	Email, Phone, Text Message

○ Classify and prioritize the incident	IT Manager/Director	Email, Phone, Text Message
	Head of IT	Email, Phone, Text Message
Notification of the incident  ○ Follow the defined IH&R plan to notify the incident	Information Security Manager	Email, Phone, Text Message
	IH&R Team	Email, Phone, Text Message
	Policy Area Lead	Email, Phone, Text Message
	Business Continuity Lead	Email, Phone, Text Message

### 3.4 Additional Information (if any)

**Note:** Refer to the following documents to fill the necessary details:

- c. Web Application Incidents Detection Template.docx
- d. Incident Identification and Validation Template.docx
- e. Incident Priority Template.docx
- f. Incident Communication Logs Template.docx
- g. Point-of-Contact Template.docx

## 4. Containment

### 4.1 Objectives

The main objective of the containment phase is to identify web applications affected by SQL injection and isolate them from other websites.

### 4.2 Containment Steps/Activities Involved

*[Activities may differ based on organizational policies, but they are not limited to the following.]*

- Activities to contain an SQL injection incident are listed below:
  - Isolate the affected database and web servers
  - Reset all affected user passwords, including administrator passwords
  - Enable the MFA mechanism to eliminate further unauthorized access
  - Abort all services provided by the web application
  - Backup all information stored in the database for forensic investigation



- Block all suspected URLs and IP addresses identified during initial investigation
- Enable the black hole feature on web applications that forces them to drop all requests from the same source after crossing a certain threshold
- If the attacks seems to originate from a single IP, administrators should blacklist or block it to prevent it from generating further traffic
- Implement Completely Automated Public Turing test to tell Computers and Humans Apart (CAPTCHA) to ensure that only humans can submit requests and forms in web applications
- Ensure that web applications do not display debugging information to users
- Maintain a backup Internet connection with a pool of IP addresses for crucial users
- Scan for injections, session-based inconsistencies, and other vulnerabilities using vulnerability scanning tools and patch them
- Find and eliminate design and coding errors in web applications
- Assuming that all input data are untrusted, whitelist, blacklist, or sanitize them according to the requirement
- Validate user input based on its length, type, format, characters, and range
- Sanitize the HTML and JavaScript code to handle untrusted inputs
- Limit access to the database based on roles (for example, underprivileged entities cannot DROP or TRUNCATE tables)
- Do not concatenate suspicious user inputs with prepared statements
- Communicate the progress:
  - Regularly inform the stakeholders about the status of the incident handling process

#### 4.3 Stakeholders Involved/Communication Established

The stakeholders involved in the aforementioned activities and their communication modes are listed below:

Activities	Stakeholders Involved	Communication Mode/Channel
Containment activities	Information Security Manager	Email, Phone, Text Message
	IH&R Team	Email, Phone, Text Message
	Resilience Lead	Email, Phone, Text Message
	Business Continuity Lead	Email, Phone, Text Message
	Policy Area Lead	Email, Phone, Text Message

#### 4.4 Additional Information (if any)

**Note:** Refer to the following documents to fill the necessary details:

- h. Containment of Web Application Incidents Checklist.docx
- i. Incident Containment Checklist.docx
- j. Incident Containment Template.docx

### 5. Analysis

#### 5.1 Objectives

The main objective of this phase is to analyze SQL injection incidents using different techniques and obtain information that can help in mitigating the incident in a timely and effective manner.

#### 5.2 Activities Involved

*[Activities may differ based on organizational policies, but they are not limited to the following.]*

- Use log analysis tools such as Loggly, Logentries, GoAccess, and Splunk to analyze the log files of IDS, web server, and database
- Use regex search to find HTML tags, SQL signature words, and their equivalents in web access logs to check for SQL injection attacks
- Use sniffing tools such as Wireshark or tcpdump to capture data packets transmitted between the attacker and webserver to detect SQL injection signatures
- Examine suspicious network traffic to obtain details such as to and from IP addresses, protocols, encryptions, and data
- Use online analysis tools such as VirusTotal to determine whether the suspected IP addresses are malicious
- Review server logs for abnormal behavior indicating SQL injection
- Examine the log files of IDS, web servers, and database for malicious SQL queries; for example:
  - `12:34:35 192.2.3.4 HEAD GET /login.asp?username=blah' or 1=1 -`
  - `12:34:35 192.2.3.4 HEAD GET /login.asp?username=blah' or )1=1 (-`  
`-`
  - `12:34:35 192.2.3.4 HEAD GET /login.asp?username=blah' or exec`  
`master..xp_cmdshell 'net user test testpass --`
- Examine network packets for signatures and related SQL injection attempts such as `1'` or `1=1 --`
- Use regular expressions to detect SQL injection meta-characters such as single quote (`'`) and double dash (`--`)

Some regular expressions used to detect SQL injection-specific characters and their meanings are listed below:

Characters	Explanation
\'	Single quote character
	Or
\%27	Hex equivalent of single quote character
\-\-	Double dash
#	Hash or pound character
\%23	Hex equivalent of hash character
i	Case-insensitive
x	Ignore white spaces in pattern
\%3D	Hex equivalent of = (equal) character
\%3B	Hex equivalent of ; (semi-colon) character
\%6F	Hex equivalent of o character
\%4F	Hex equivalent of O character
\%72	Hex equivalent of r character
\%52	Hex equivalent of R character

- Review the affected database to determine if it contains critical data and the amount of data stolen

### 5.3 Stakeholders Involved/Communication Established

The stakeholders involved in the aforementioned activities and their communication modes are listed below:

Activities	Stakeholders Involved	Communication Mode/Channel
Initiate evidence gathering and forensic analysis	Information Security Manager	Email, Phone, Text Message
	IH&R Team	Email, Phone, Text Message
Analyze the scope of SQL injection incident	CISO	Email, Phone, Text Message
	Information Security Manager	Email, Phone, Text Message

	IT Manager/Director	Email, Phone, Text Message
	IH&R Team	Email, Phone, Text Message
Analyze the log files of IDS, web servers, and database ○ Use log analysis tools ○ Use regular expressions to detect SQL injection meta-characters	Information Security Manager	Email, Phone, Text Message
	IH&R Team	Email, Phone, Text Message
	Resilience Lead	Email, Phone, Text Message
	Business Continuity Lead	Email, Phone, Text Message
	Policy Area Lead	Email, Phone, Text Message

#### 5.4 Additional Information (if any)

**Note:** Refer to the following documents to fill the necessary details:

- k. Web Application Incident Analysis Template.docx
- l. Web Application Incident Handling Toolkit.docx
- m. Checklist for Handling the Forensic Evidence Properly.docx
- n. Evidence Gathering and Forensic Analysis Form.docx

## 6. Eradication

### 6.1 Objectives

The main objective of this phase is to eliminate the root cause of SQL injection incident and take appropriate measures to prevent recurrence.

### 6.2 Eradication Steps/Activities Involved

*[Activities may differ based on organizational policies, but they are not limited to the following.]*

- Activities performed to eradicate SQL injection incident:
  - Sanitize and filter user inputs, analyze the source code for SQL injection, and minimize the use of third-party applications
  - Establish a connection using a non-privileged account and grant the least privileges to the database, tables, and columns to avoid SQL injection on the database
  - Limit the length of user inputs, which can act as a defense-in-depth security measure to filter integers and variables

- Use custom error messages created by developers, which do not reveal significant information to attackers
- Monitor database server traffic by implementing IDS or WAF for SQL injection requests to the database
- Disable commands such as xp\_cmdshell that run OS commands in the context of SQL server accounts
- Always use a method attribute set to POST to prevent SQL injection attacks on the web server
- Run database service accounts with minimal rights to reduce the risk of data manipulation
- Move extended stored procedures included in SQL servers to an isolated server
- Ensure that the server only accepts alphanumeric characters as the username using functions such as IsNumeric() and type-safety
- Implement server-side validation to avoid the execution of raw HTTP calls
- Implement the object relational mapping (ORM) framework that converts tables into objects, which can be used for safe communication
- Employ character escaping to escape special characters such as “/, -”
- Implement controls such as LIMIT within queries to minimize the number of records revealed to users
- Use dynamic application security testing (DAST) tools such as Invicti, Acunetix Vulnerability Scanner, and HCL App to detect SQL injection flaws in web applications
- Use static application security testing (SAST) tools such as Codacy, Appknox, AttackFlow, and bugScout to detect and remove vulnerabilities in the application code
- Set up time-to-live (TTL) settings to optimize the page load
- Design access controls for the website login page
- Reset the password of web application administration and management accounts
- Block associated indicators such as URLs, domains, IP addresses, and file hashes in security controls
- Implement ORM engine to map application code objects to the relational database
- Use ApexSQL Operations Toolkit for the SQL Server toolkit to deal with performance issues and other security vulnerabilities

- Use solutions such as AutosyncDB and ApexSQL Compare & Sync Toolkit to compare and synchronize SQL server database schemas with predefined data
- Review the identified vulnerabilities with the developers and guide them on secure coding practices

### 6.3 Stakeholders Involved/Communication Established

The stakeholders involved in the aforementioned activities and their communication modes are listed below:

Activities	Stakeholders Involved	Communication Mode/Channel
Develop an eradication plan ○ Perform technical and business analyses and create a prioritized eradication plan ○ Establish a communication strategy based on the eradication plan	Information Security Manager	Email, Phone, Text Message
	IH&R Team	Email, Phone, Text Message
	IT Manager/Director	Email, Phone, Text Message
	Internal/External Communications Team	Email, Phone, Text Message
	Resilience Lead	Email, Phone, Text Message
	Business Continuity Lead	Email, Phone, Text Message
	Policy Area Lead	Email, Phone, Text Message
Eradication activities	Information Security Manager	Email, Phone, Text Message
	IH&R Team	Email, Phone, Text Message

### 6.4 Additional Information (if any)

**Note:** Refer to the following documents to fill the necessary details:

- o. Eradication of Web Application Incidents Checklist.docx
- p. Incident Eradication Template.docx
- q. Incident Eradication Checklist.docx

## 7. Recovery

### 7.1 Objectives

The main objective of this phase is to restore web application services to the normal operational state and maintain business continuity.

## 7.2 Recovery Steps/Activities

*[Activities may differ based on organizational policies, but they are not limited to the following.]*

- Activities to recover from SQL injection incident:
  - Restore web application functionalities from the backup application code
  - Restore web servers and databases from clean and trusted backups
  - Elevate the logging and monitoring levels of the web application to gather realistic information about the latest events
  - Check web application backups for traces of the attack and clean them
  - Change the administrative passwords of all devices and resources
  - Patch the exploited vulnerabilities and test them under different scenarios
  - Examine the current web application code and ensure that all flaws are removed before the release
  - Rebuild the entire system if backup is not available for the damaged systems
  - Check whether the application has recovered completely, along with user accounts, privileges, and configurations
  - Use an access control matrix and define access control rules with a list of accessible and authorized requests
  - Continue monitoring the web application for a while longer after completely recovering the system from the incident
  - Reconfigure the IDS for faster detection of similar incidents in future
  - Improve the security of the network perimeter by implementing strict WAF, IDS, and ACL policies and rules
  - Use tools such as ApexSQL Log and SysTools SQL Recovery to recover from SQL injection incidents

## 7.3 Stakeholders Involved/Communication Established

The stakeholders involved in the aforementioned activities and their communication modes are listed below:

Activities	Stakeholders Involved	Communication Mode/Channel
Recover activities	Information Security Manager	Email, Phone, Text Message
	IH&R Team	Email, Phone, Text Message

## 7.4 Additional Information (if any)

**Note:** Refer to the following documents to fill the necessary details:

- r. Recovery of Web Application Security Incidents Checklist.docx
- s. Incident Recovery Procedure Template.docx
- t. Incident Recovery Checklist.docx

## 8. Post-incident Activities

### 8.1 Objectives

The main objective of this phase is to create the necessary SQL injection incident reports such as incident documentation, lessons learned, and incident impact assessment. Another objective of this phase is to officially close the SQL injection investigation and disclose its details to respective stakeholders.

### 8.2 Activities Involved

*[Activities may differ based on organizational policies, but they are not limited to the following.]*

- Create incident post-mortem, root cause analysis (RCA), and incident review reports to understand the determination process of the root cause of SQL injection incident; this will help in learning from failures and implementing effective measures to prevent similar incidents in future
- Create an after action report (AAR) that includes information such as what worked effectively, areas of improvement, and strategies for enhancing the response in case of similar SQL injection incidents
- Conduct a lessons learned meeting to document the details of the SQL injection incident; ensure that the following questions are answered in this meeting:
  - When and who detected the SQL injection incident?
  - What happened exactly?
  - What caused the SQL injection incident?
  - To whom was the SQL injection incident reported?
  - What were the challenges encountered during the incident handling process?
  - Was the organization adequately prepared for the SQL injection incident?
  - How was the SQL injection incident contained?
  - How were the impacted systems sanitized?
  - Were the existing tools effective during the response process?
  - What procedures were followed during recovery?
  - Were the documented procedures followed by the response team?



- How well did the incident response team and management perform in resolving the SQL injection incident?
- How should the incident response team and management respond to mitigate similar incidents in future?
- Were there any gaps in communicating the SQL injection incident?
- Was the right amount of information shared with the right personnel?
- What tools and resources are required to detect, analyze, and prevent SQL injection incidents in future?
- Create concise and clear SQL injection incident documentation in a standard format and get it reviewed by an editor
- Create an incident impact assessment report to determine all types of losses caused by the SQL injection incident; this report must address the following, if required:
  - Financial losses caused by the SQL injection attack
  - Legal costs for investigating the case, lawyer's fees, etc.
  - Costs pertaining to analyzing the SQL injection incident, including recovery and installation of software and hardware
  - Implementation costs
  - Costs related to the damage of goodwill as well as loss of customer trust and reputation
- Officially close the SQL injection investigation by informing the management and securely retain investigation reports considering the retention policy of the organization
- Review the document with subject matter experts (SMEs) for further improvement
- Disclose incident details to the respective stakeholders by consulting with the legal department of the organization

### 8.3 Stakeholders Involved/Communication Established

The stakeholders involved in the aforementioned activities and their communication modes are listed below:

Activities	Stakeholders Involved	Communication Mode/Channel
Conduct lessons learned meeting	Information Security Manager	Email, Phone, Text Message

	IH&R Team	Email, Phone, Text Message
Create incident documentation	Information Security Manager	Email, Phone, Text Message
	IH&R Team	Email, Phone, Text Message
Create an incident impact assessment report	Information Security Manager	Email, Phone, Text Message
	Incident Response Team	Email, Phone, Text Message
Close the investigation officially	Information Security Manager	Email, Phone, Text Message
	IH&R Team	Email, Phone, Text Message
	Senior Management	Email, Phone, Text Message
Disclose incident details to the respective stakeholders	Information Security Manager	Email, Phone, Text Message
	Manager - Information Governance	Email, Phone, Text Message
	IT Manager/Director	Email, Phone, Text Message
	CISO	Email, Phone, Text Message
	Legal Team	Email, Phone, Text Message
	Human Resource	Email, Phone, Text Message
	Media	Email, Phone, Text Message
	Vendors	Email, Phone, Text Message
	Customers and General Public	Email, Phone, Text Message

	Business Partners	Email, Phone, Text Message
	Resilience Lead	Email, Phone, Text Message
	Business Continuity Lead	Email, Phone, Text Message
	Policy Area Lead	Email, Phone, Text Message

#### 8.4 Additional Information (if any)

**Note:** Refer to the following documents to fill the necessary details:

- u. Incident Documentation Template.docx
- v. Incident Impact Assessment Report Template.docx
- w. Incident Closure Letter.docx
- x. Incident Disclosure Form.docx

#### 9. Appendix (if any)