



**Northeastern
University**

**Incident Response Playbook
(Covering 3 attack scenarios)**

(Initial Draft version to share as Work Sample)

Name: Davindernain Singh Sehgal

Prof: Jose Sierra

Scenarios: 1, 4 and 10

Date: Dec 1, 2021

Abstract

This Incident Response playbook is a collection of 3 total scenarios that a CSIRT should take against an incident to prevent the worst-case scenario in any case. The different scenarios are:

1. Unauthorized attempt to access Payroll records
2. SQL injection vulnerability
3. Zoom link leaked

All these scenarios are difficult to handle in themselves, so dividing a well-informed and well-formed team is no less than a blessing in this situation. The most difficult part of the study was to put myself in the shoes of the incident response managers' shoe. In a situation where an attack suddenly happens, it can get very pressurizing for the team, and working under stress or pressure does not create good and right ideas on time, this is where my playbook will be useful because all the steps have been simply stated in it that the team can refer and can make use of to perform the right steps.

Most incidents have never been studied before and I had to personally research and read on topics, this helped me learn a lot from this experience, and in the future, I would like to get the opportunity to run this through on a table test/ table read with a team and make these responses better.

Table of contents

Sr.no	Title	Page no
1.	Scenario 1	4
2.	Scenario 4	7
3.	Scenario 10	9
4.	Reference	13

Scenario 1 Purpose: Unauthorized attempt to access payroll

Identification/Detection :

1. Unexplained system failures or restarts
 - Checking the logs for system failures or restarts in the recent times
2. Access to systems outside of normal business hours
 - It can be assumed that the attacker may try to access the system when the actual user is not using the same credentials, looking at the logs for login times will be useable information.
3. Logins on multiple systems using a specific account
 - Looking at the logs to find out the number of different Mac addresses used to access the particular login id
4. Unexplained browsing of records
 - Interviewing the concerned user in question to find out why they were browsing a particular record regardless of whether they were privileged to view that or not.
5. Increased logins to a computer system using that account
 - Find out the account used to log in the most number of times.
6. The actual user of the account is unable to login into his account
 - Check if the actual owner of a particular account is able to gain access to their account or not.
7. Simultaneous logins on the same account via two different machines
 - The server gives an alert when a user accesses the system from two different machines at the same time.

2 Notification:

1. The first step to be taken should be to call and inform the CSIRT As soon as possible.
2. The CSIRT team will keep their legal advisor on standby for legal and ethical advice that might be needed during the investigation.
3. CSIRT team will then first report this incident to the Incident Response manager and then Technology and operations team lead will be notified about the same.

3. Analysis:

1. Check if other records were attempted to access
 - There are chances that an attacker might have tried to access other records using the compromised account
2. Interviewing the employee at the help desk who was used for the social engineering attack:

- Do they remember any of the conversations that took place?
 - Was the call recorded?
 - Ask for the mobile number from which the call came.
3. Identify the internal aspects of this incident
 - Was the caller able to correctly answer the questions that only the original user of the account should have known?
 - If yes? Who had access to those records?
 - If no? Why did the helpdesk not notify us that such an incident occurred before?
 4. Checking logs:
 - Check the logs for the last time the compromised user-id was used to get into the system
 - Check the logs for the files accessed by the compromised user-id
 - Check the logs for any file transferred by the compromised user-id
 5. Identifying additional technical risks:
 - Check if any other social engineering attack has happened in the same format for any other account.
 - Check if any other attack has been attempted on the account.
 - Check for other unusual login activities.
 6. Confirm that this issue is flagged and so is the account in question, if not, manually flag the account.
 7. Collect all the evidence related to this case using SIEM.

4. Containment:

1. Identify the systems that have been affected:
 - If the Servers have been compromised:
 - > Take the latest backup of data into an external source immediately
 - > Retract all privileges given to the account in question.
2. Identify if other systems that can be accessed using the same credentials:
 - Immediately block access to those systems for that account.
 - Put up an alert on these systems to flag a connection attempt from the account in question.
3. Check the Intrusion Prevention Sensors on the internal network and enforce them stronger than before.
4. Identify any source attribution collected :
 - Check for missing database files
 - Check the vulnerability logs
 - Check the system logs

5. Identify the tools used to detect the attack:

- SIEM
- Scanners
- Firewall
- IDS
- Or Access Control Systems (will report when a user tried to access a file that they aren't given the privilege to open)

5. Recovery:

1. Update:

- Update IDS with the latest patches
- Updates IPS with the latest patches
- Update the Firewall with the latest patches

2. Identify ways to stop the same attack in the future:

- Enable 2FA(@ factor authentication)
- Let the users choose their own Security questions that only they can correctly answer

3. If possible take a backup of the server and database and wipe it up to remove potential malware that an attacker might have added to the system for backdoor access.

4. Report the whole incident and inform the employees who handle customers about how to approach and prevent social engineering Attacks

Scenario 4 Purpose: SQL injection vulnerability

1. Identification/Detection:

1. Server logs flag a malicious SQL fragment that they were not expecting to come through.
2. The firewall detects SQL queries instead of HTTP responses.
3. Regular audits are conducted within the company or by a hired external source.
4. Application/ Database checks to find our vulnerabilities.
5. Has a vulnerability already been exploited?
 - If Yes, what was lost, and how to protect the rest of the data?
 - If Not, How was this vulnerability left unseen, and how to prevent it from getting exploited by someone?

2. Notification:

1. Notify the CSIRT team immediately after finding out.
2. The Incident Response manager then calls in the security analysts, the CISO and the technology and operations team lead.
3. The senior management should only be informed about this.

3. Analysis:

1. Identify the level of risk for this particular threat.
 - Mild risk: Concerning Vulnerability, action can be taken if needed
 - Moderate Risk: the Vulnerability involves repeated disruption, and action should be taken to mitigate it.
 - Severe Risk: Disrupted vulnerability, Mitigation should begin as soon as one can.
 - Extreme Risk: Way off Baseline vulnerability, the team should start working on it the very instant it is discovered.
2. What was the target? What part of the server was the command trying to exploit?
3. Was the request made internally using privileges or was it made externally?
 - If the attack was done internally using a set of privileges, it could either mean that another attack has already taken place or this is an inside job.
 - If the attack was done externally, it is important to look at the logs and find out as much as one can about the particular attacker.
4. What was the attacker trying to achieve? This could be found by looking at the logs and finding out if the HTTP request was GET or POST.
5. Find out what happens when that vulnerability is exploited, and check if the same instance is caused in the past.
 - If yes, when and by whom?

- If not, check how the vulnerability got leaked to the attacker.

Containment:

1. Begin Input validation across all servers, where a user will only be able to input data that is valid. This will reduce the chances of SQL injection attacks as the servers will not accept an SQL query in the form of an input.
2. Implement the stored procedure
 - Separate the database from user inputs with separate commands, so that even if someone uses an SQL injection attack, it reduces the chances of it working.
3. Immediate enforcement of the principle of least privilege.
 - Doing this would immediately restrict unauthorized users from making any changes to the database.
 - Only users given the privileges can access the data and only use a specified gateway.
4. Mitigate The impact of SQL injection vulnerabilities by enforcing the least privilege on the database.

Recovery:

1. Add this vulnerability that has been discovered to the Vulnerability patching program under the Incident response manager's supervision.
2. Validate the Escape input procedure
 - Specific characters that are used constantly in SQL queries such as “ ‘ ”, shouldn't be allowed to be entered by a user.
3. Add alerts in the firewalls and servers so that if another vulnerability amongst the same lines is ever tried to exploit, the admins will get notified immediately.

Scenario 10 Purpose: Zoom Link Leaked

1. Identification/ Detection:

1. How was the link leaked:
 - Did someone share it via an unsecured channel?
 - > Was it done purposely or by mistake?
2. Interview the Employee who shared the link in the wrong Whatsapp group:
 - Were the names of the group similar on any level?
 - When did they realize that they have made a mistake?
 - Why did they not retract the message?
 - Why did they not inform any superior about the link being shared with the wrong group?
3. How was the incident brought to attention:
 - Was it before or after an unidentified individual joined the meeting?

2. Notification :

1. A member of the CSIRT team will be notified first who will then inform the Incident Response manager.
2. The Incident Response manager will then bring forth his Security analysts
3. The legal advisor will be kept on standby, as they might be needed
4. Public relations officer might be contacted to generate a statement that might be needed to publish so that discussions in the meeting do not leak.
5. The host of the meeting notices an uninvited user in the meeting and then reports it to CSIRT.

3. Analysis:

1. Identifying Human error:
 - It is feasible that the leak took place because of a genuine mistake made by an employee.
2. Checking Zoom meetings logs:
 - Check for the logs to see if you can find the unidentified user who logged in Zoom meeting.
 - Find out if you can see the email address of the user who joined in.
 - Check if the same user has joined any other company meeting in the past.
3. Read the transcript of the Zoom meeting:
 - See if sensitive information was shared in the Zoom meeting or not.

- Check the time when the user joined in to find out if they have witnessed the sharing of sensitive data
4. Match the username or email id of the unidentified individual to anyone on the WhatsApp group, in which the link was shared.

4. Containment:

1. Strategies:
 - If strategies were discussed during this meeting, it is feasible to change the strategies or simply do them at their earliest.
2. After we find out how much the intruder has listened to during the meeting it is possible to work out a plan for future mitigation
 - If the intruder has not listened to important and sensitive information, the team can go ahead with the plan they discussed during the meeting.
3. Discussion with the human resources department and senior management about this incident can prove beneficial.
 - The Human resources department has data on the current as well as ex-employees this can be used to reach out to the intruder if their records from HR match with the credentials from Zoom logs
4. Create a new secure communication medium where the employees won't make a mistake.
5. Keep an eye out for moves made by the competitor company to find out if someone stole ideas or information from the meeting and if the whole meeting has now been leaked.

5. Recovery:

1. If an incident like this occurs, it is very important that an official group must be created with only the current employees.
2. 2FA(2-factor authentication) should be enabled for zoom meetings so that only a person with a valid company Email address gets to join the meeting.
3. Inform all members of the meeting about the intruder in the meeting that was conducted.
4. Enable the "waiting room feature" for Zoom meetings so that the host and admin can monitor who all are joining the Zoom virtual meeting.
5. Once the meeting begins, the host/admin can lock the meeting for any outsiders to join.
6. Always use a random meeting ID and send/broadcast it personally to all the members who should be present in the meeting.

References:

- <https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-184.pdf>
- <https://www.edureka.co/blog/sql-injection-attack#PreventingSQLInjection>
- <https://www.trendmicro.com/vinfo/us/security/definition/indicators-of-compromise>
- <https://www.extrahop.com/company/blog/2020/sqli-attacks-definition-and-how-to-protect-against-them/>
- <https://csrc.nist.gov/publications/detail/sp/800-61/rev-2/final>
- <https://www.csoonline.com/article/3257429/what-is-sql-injection-how-these-attacks-work-and-how-to-prevent-them.html>
- <https://www.cnet.com/tech/services-and-software/how-to-prevent-zoombombing-in-your-video-chats-in-4-easy-steps/>