

**Department of Engineering Technology**

SET-222

Software Operations & Maintenance

Experiment # 12

**Experiment Title**

Incident Response and Root Cause Analysis

**Assessment of CLO(s): 03**

**Performed on \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

|  |  |  |  |
| --- | --- | --- | --- |
| **Student Name:** |  | | |
| **Roll No.** |  | **Group** |  |
| **Semester** |  | **Session** |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **S. No.** | **Perf. Level**  **Criteria** | **Excellent**  **(2.5)** | **Good**  **(2)** | **Satisfactory**  **(1.5)** | **Needs Improvement**  **(0 ~ 1)** | **Marks Obtained** |
| **1** | Project Execution & Implementation | Fully functional, optimized, and well-structured. | Minor errors, mostly functional. | Some errors, requires guidance. | Major errors, non-functional, or not Performed. |  |
| **2** | Results & Debugging  Or Troubleshooting | Accurate results with effective debugging  Or Troubleshooting. | Mostly correct, some debugging Or Troubleshooting needed. | Partial results, minimal debugging  Or Troubleshooting. | Incorrect results, no debugging Or Troubleshooting, or not attempted. |  |
| **3** | Problem-Solving & Adaptability  (VIVA) | Creative approach, efficiently solves challenges. | Adapts well, minor struggles. | Some adaptability, needs guidance. | Lacks innovation or no innovation, unable to solve problems. |  |
| **4** | Report Quality & Documentation | Clear, structured, with detailed visuals. | Mostly clear, minor gaps. | Some clarity issues, missing details. | Poorly structured, lacks clarity, or not submitted. |  |
| **Total Marks Obtained Out of 10** | | | | | |  |

**Experiment evaluated by**

|  |  |  |  |
| --- | --- | --- | --- |
| **Instructor’s Name** | **Ms. Shagufta Aftab** | | |
| **Date** |  | **Signature** |  |

## Copyright © Department of Engineering & Technology – UIT University Karachi

**Objective:**

* To understand the key phases of incident response.
* To simulate an incident response process using a real-world scenario.
* To perform root cause analysis (RCA) on the identified security incident.
* To document findings and recommend preventative measures.

**3. Prerequisites:**

* Basic knowledge of cybersecurity principles.
* Familiarity with network and system logs.
* Understanding of operating system (Linux/Windows) commands.
* Tools: Wireshark, Event Viewer, Splunk (optional), ELK Stack (optional), or any log analysis tool.

**4. Tools & Software Required:**

* Virtual Machine (Kali Linux, Windows)
* Wireshark
* Sysinternals Suite / Event Viewer
* Text Editor (Notepad++, Sublime, etc.)
* Internet connection for updates/tools

**5. Lab Setup:**

1. Prepare two virtual machines:
   * **Attacker Machine** (Kali Linux)
   * **Victim Machine** (Windows or Ubuntu)
2. Ensure logging is enabled on the victim machine:
   * System Logs
   * Security Logs
   * Application Logs
3. Set up basic services (e.g., SSH, HTTP server) for exploitation.

**6. Lab Procedure:**

**Part A: Simulating an Incident**

1. **Initiate an Attack**
   * Use a basic scanning tool (e.g., Nmap) from the attacker machine to discover open ports on the victim machine.
   * Perform a brute force or SQL injection attack using tools like Hydra or SQLMap.
2. **Detect the Incident**
   * Monitor logs on the victim machine.
   * Capture network traffic using Wireshark or tcpdump.
   * Identify suspicious activity (failed logins, unusual traffic, etc.).

**Part B: Incident Response Process**

1. **Preparation**
   * Confirm logging is functioning.
   * Define roles/responsibilities if working in a team.
2. **Identification**
   * Review logs to detect abnormal behavior.
   * Identify the type, scope, and impact of the incident.
3. **Containment**
   * Isolate the affected system or network segment.
   * Disable suspicious services or block IPs.
4. **Eradication**
   * Remove malicious files/scripts.
   * Patch vulnerabilities.
5. **Recovery**
   * Restore from backups if needed.
   * Monitor systems to ensure normal operation.
6. **Lessons Learned**
   * Document the timeline and findings.
   * Conduct a post-mortem with the team.

**Part C: Root Cause Analysis (RCA)**

1. **Define the Problem**
   * What happened? When? What was the impact?
2. **Collect Data**
   * Review system logs, firewall logs, and network captures.
   * Interview users or administrators if needed.
3. **Identify Root Cause**
   * Use methods like:
     + **5 Whys**
     + **Fishbone Diagram (Ishikawa)**
     + **Timeline Analysis**
4. **Implement Corrective Actions**
   * Patch software
   * Strengthen firewall rules
   * Update incident response playbooks

**7. Observations and Screenshots**

* Include screenshots of:
  + Detected log entries
  + Network traffic from Wireshark
  + Attack evidence (e.g., brute force attempts)

**8. Results:**

Document:

* What was the incident?
* How was it detected?
* What was the root cause?
* What actions were taken to mitigate and prevent recurrence?

**9. Conclusion:**

Summarize key learnings:

* Importance of structured incident response
* Value of RCA in preventing repeat incidents
* Real-world significance in a cybersecurity operations center (SOC)

**10. Viva Questions (Oral Evaluation):**

1. What are the six phases of the incident response lifecycle?
2. How does containment differ from eradication?
3. What is the purpose of root cause analysis?
4. What tools can be used for RCA?
5. Explain the 5 Whys technique.