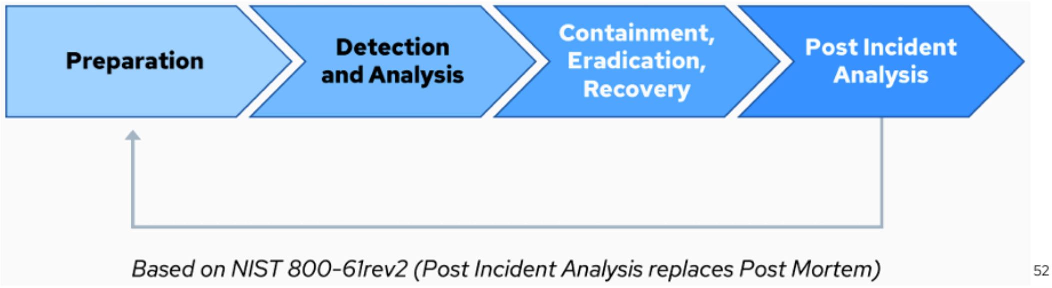
Domain 11 Incident Response & Resilience

Incident Response:

- An'**event**' is any observable issue on a cloud platform that may indicate an underlying security or availability = can escalate to an 'incident' when it is determined to violate security policies.
- **Incidents** require immediate attention to contain and mitigate their effects.
- '**breaches**,' which signify a successful penetration or circumvention of security measures.



Based on NIST 800-61rev2 (Post Incident Analysis replaces Post Mortem)

Figure 29: Phases of IR Life Cycle in Cloud Security

Preparation: Establish an incident response capability to respond to incidents. This entails the following:

- Establish an incident response process
- Build a team and assign roles and responsibilities
- Train the team and run exercises
- Establish a communication plan and facilities
- Responder access to environments
- Responder access to tools: incident analysis services, hardware, and software
- Internal documentation (e.g., port lists, asset lists, network traffic baseline)
- Evaluate infrastructure: proactive scanning and monitoring, vulnerability and risk assessments
- Subscribe to third-party threat intelligence services.

Conduct backup restoration testing regularly and disaster recovery (DR) tests at least once per year to ensure that incident response plans are up-to-date and effective.

Detection&Analysis: Identify security incidents and analyze their impact. This entails the following:

- Detection engineering
- Alerts: This includes Cloud Security Posture Management (CSPM), security information and event management (SIEM), workload protection, and network security monitoring.
- Validate alerts (reduce false positives), with escalation.

● Estimate the scope of the incident.
● Assign an Incident Manager to coordinate actions.
● Build a timeline of the attack.
● Determine the extent of the potential data loss or impact.
● Notify and coordinate activities.
● Communicate the incident containment and recovery status to senior management.

Containment,Eradication, &Recovery: Isolate the incident to prevent further damage and remove the root cause; recovery: restore affected systems.
● Containment: Isolate identities and workloads, taking systems or services offline, and consider the trade offs between data loss versus service availability.
● Eradication & Recovery: Clean up compromised assets and restore systems and services to normal operation. Deploy controls to prevent similar incidents.
● Document the incident and gather forensic evidence (e.g., chain of custody).

Post-Incident Analysis : Learn from the incident, document, and improve future responses.
● Lessons learned: Which detections worked, and which alerts fired properly? What detections and protections need to be created based on the event? What improvements does the incident response process need to make? What Indicators of Compromise (IOC) were discovered and were they shared with the community?

Preparation:

- Cloud incidents are shared incidents, even when the customer owns all of the affected resources.
- The cloud incident response team should have persistent read access to all deployments. (All use of these privileges should be logged and reviewed.) (Readaccesstometadataandconfigurations, sometimes called “security audit,” + not just the metadata, can and in many cases should require multiple approvals to use and follow a “break glass” process.)

Detection & Analysis:

- There are multiple places to perform detection as seen :
- CDR ● SIEM ● CSPM / Cloud Native Application Protection Platform (CNAPP) ● CSP/Identity Provider (IdP).

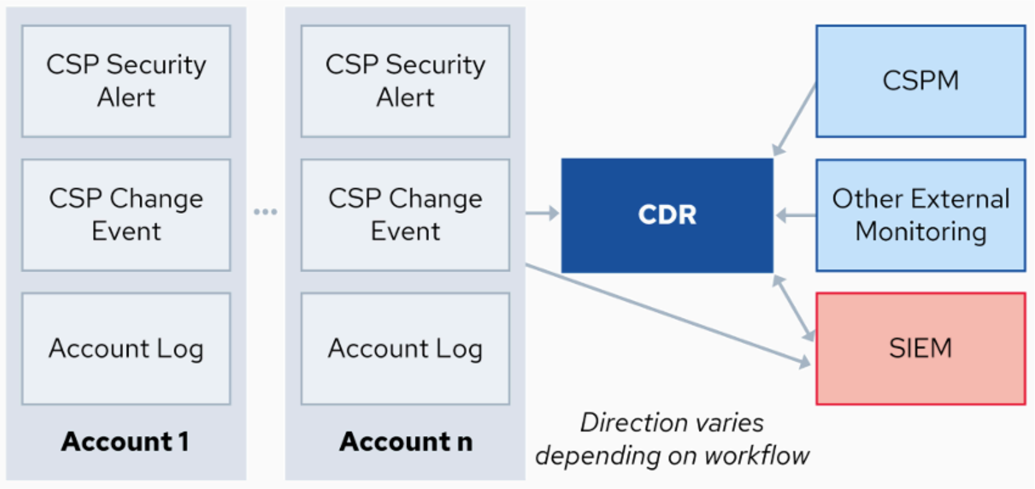


Figure 30: Figure 30: Example of an IR Analysis Workflow

Cloud System Forensics:

- Snapshots: all CSPs and container management systems support snapshots.
- Volatile memory acquisition: the responders will need to install software tools which will also affect the system.
- Log analysis
- Evidence preservation : understanding of the backup and data retention policies of both the CSP and the CSC and the chain of custody of snapshots.
- Containers:
  - Are naturally ephemeral, often existing for only short periods. = challenging forensics.
  - Capturing container logs and snapshots of container states to provide insights = redirect container logs, VM logs, and service logs to external log storage.
- Serverless computing

Containment:

- IAM and management plane containment should be the top priorities in any security incident = can be very difficult.
- Understanding of whether the attacker was able to use their access to escalate or pivot into different identities, just as we track attackers pivoting around a network.
- often **easier** in **cloud** networks since it relies on Software Defined Networking.
- For critical data, containment may need to risk breaking application functionality temporarily, and incident responders should have a timely escalation path to the authority and capability to make this decision and take action in highly critical situations.
  Eradication:
- remove the attacker from the management plane = credential rotation, adding additional policy conditions, adding MFA or digital certificates, and similar techniques.

- often requires deleting old versions of images, serverless code, and IaC (Attackers may use these to re-compromise a deployment)
  Recovery
- IaC, autoscaling, and other automation are incredibly powerful for incident recovery.
- should be analyzed to ensure that the root cause was eliminated.

Post Incident Analysis:

- Responders should be required to create a new runbook /playbook for new incident types they encounter.
- Rather than assigning blame, CSA recommends following a Just Culture approach, which focuses on the identification of systemic failures before blaming individuals, while still holding individuals accountable for their actions.
  For example, if over-privileged IAM was the source of the breach, the CSC may consider utilizing scanners to identify potential IAM issues. Security may provide common baselines and work with teams to review permissions. Alternatively, the organization can move from static credentials to Just-in-Time entitlements combined with strong authentication, using frictionless tooling that doesn’t slow down developers.