CYBERSECURITY ATTACKS PLAYBOOKS RELATED TO AMAZON WEB SERVICES (AWS)

BY IZZMIER IZZUDDIN

TABLE OF CONTENTS

AWS PLAYBOOK: RESPONDING TO UNUSUAL EC2 INSTANCE LAUNCHES

1. PREPARATION

Baseline Security Configurations

- **Set Up GuardDuty**:
  - Enable across all accounts and regions to detect anomalous EC2 activity.
- **IAM Policies**:
  - Limit who can launch EC2 instances using restrictive IAM roles and permissions.
  - Implement Service Control Policies (SCPs) in AWS Organisations to restrict regions or instance types.
- **CloudTrail Logging**:
  - Ensure AWS CloudTrail is enabled for all regions to track EC2 launch events.
  - Store logs in an encrypted, private S3 bucket for analysis.

Monitoring and Alerts

- **Use Amazon CloudWatch to:**
  - Set alarms for unexpected instance launches in restricted regions.
  - Detect usage of non-standard AMIs or configurations.

Tagging Standards

- Enforce consistent tagging (e.g., Owner, Environment) for all instances to identify legitimate instances easily.

Training and Awareness

- Train administrators to recognise suspicious EC2 activity and follow incident response procedures.
- Conduct regular simulation exercises, including scenarios of unauthorised instance launches.

2. DETECT

Anomalous Launch Indicators

- **GuardDuty Findings**:
  - “EC2 instance behaving as a botnet controller.”
  - “Unusual ports or geolocations detected in instance network traffic.”
- **CloudTrail Logs**:
  - Review RunInstances events for:

        - Unexpected user agents or IP addresses.
        - Launches in unapproved regions.
- **CloudWatch Metrics**:
    - Spikes in EC2 resource utilisation (e.g., unexpected CPU or network activity).

Potential Triggers

- **Unusual Instance Behaviour**:
    - Instances launching with AMIs that aren't pre-approved.
    - Unfamiliar key pairs or security groups attached to instances.
- **Unexpected Costs**:
    - Sudden increases in billing due to unauthorised instance launches.

3. ANALYSE

Review Logs

- **CloudTrail**:
    - Identify who initiated the instance launch and their IP address.
    - Check for changes to permissions or credentials before the event.
- **VPC Flow Logs**:
    - Look for unusual outbound traffic from the instance.
- **Instance Metadata**:
    - Verify instance details such as AMI ID, instance type and tags.

Categorise the Incident

- Was the launch caused by:
    - Credential compromise?
    - Misconfigured IAM policies?
    - Malicious insiders?

Assess Impact

- Check for:
    - Instances communicating with known malicious IPs or domains.
    - Data exfiltration or lateral movement from the instance.

4. CONTAIN / ERADICATE

Containment Steps

- Isolate the instance:
    - Move the EC2 instance to a quarantine VPC or security group.

    - Disable internet access by modifying route tables or NAT gateway.
- Block malicious actions:
    - Revoke credentials used to launch the instance.
    - Update security groups to block unexpected traffic.

Eradication Steps

- Terminate the unauthorised instance.
- Review and update IAM roles/policies that allowed the launch.
- Scan the compromised account for additional unauthorised activity.

5. RECOVER

Validate Recovery

- Confirm that no unauthorised EC2 instances remain in the environment.
- Ensure all IAM credentials have been rotated if compromised.

Restore Configurations

- Reinforce tagging policies and access restrictions for EC2.
- Update CloudTrail filters to detect similar events earlier.

6. LESSONS LEARNT

Post-Incident Review

- Analyse the root cause of the incident:
    - Weak access control?
    - Missing alerts or monitoring gaps?
- Update playbooks to include new detection and response measures.

Enhanced Security Practices

- Automate detection with tools like AWS Lambda.
- Increase frequency of IAM policy audits to prevent privilege escalation.

AWS PLAYBOOK: RESPONDING TO UNAUTHORISED IAM POLICY CHANGES

1. PREPARATION

Baseline Security Measures

- **Enable AWS CloudTrail**:
  - Log all management events, including PutPolicy, AttachPolicy and UpdatePolicy.
  - Configure logs to route to a secure, centralised S3 bucket.
- **Set Up AWS Config**:
  - Ensure AWS Config tracks changes to IAM policies and generates alerts for non-compliant configurations.
- **Enforce Least Privilege**:
  - Use restrictive IAM policies to limit who can modify IAM roles and policies.
  - Implement SCPs in AWS Organisations to prevent high-risk changes globally.

Monitoring and Alerts

- **Use Amazon CloudWatch and AWS Config Rules to:**
  - Monitor for updates to AdministratorAccess or overly permissive policies (*:*).
  - Alert on any changes to critical roles or high-privilege accounts.

Auditing and Simulation

- Conduct regular audits of IAM roles, policies and permissions.
- Use IAM Access Analyser to simulate and identify potential unintended access paths.

2. DETECT

Indicators of IAM Policy Changes

- **CloudTrail Events**:
  - Look for events like PutPolicy, AttachUserPolicy or UpdateAssumeRolePolicy.
  - Check the actor's IP address and access method (console, CLI, SDK).
- **AWS Config Notifications**:
  - Detect non-compliant policy changes (e.g., policies granting s3:* or iam:* to unauthorised users).
- **CloudWatch Alerts**:
  - Spike in failed or successful access attempts following policy changes.

Anomalous Activity

- Sudden changes to critical roles (e.g., roles used for CI/CD pipelines or administrative tasks).
- New policies granting excessive privileges (Action: "*" or Resource: "*") to non-admin users.

3. ANALYSE

Review Logs and Changes

- **CloudTrail**:
  - Identify the user or role that made the change, time of the event and originating IP.
- **IAM Policy History**:
  - Compare the previous version of the policy with the modified version.
- **AWS Config Timeline**:
  - Check for sequential events to understand the scope of changes.

Assess Impact

- Determine if the changes:
  - Enable privilege escalation (e.g., granting iam:PassRole to a low-privilege user).
  - Provide unauthorised access to sensitive data or services.
  - Increase risk of data exfiltration or unauthorised deployments.

4. CONTAIN / ERADICATE

Containment Steps

- Revert unauthorised policy changes:
  - Use AWS CLI or Management Console to restore the original policy.
- Temporarily restrict access:
  - Disable the user or role that initiated the change.
  - Apply SCPs or deny rules to prevent further modifications.

Eradication Steps

- Rotate all potentially compromised credentials (e.g., access keys, session tokens).
- Revoke permissions for users or roles associated with the policy change.
- Update monitoring rules to detect similar changes in the future.

5. RECOVER

Restore Secure State

- Reapply IAM policy baselines to ensure all roles and users are compliant.
- Validate policy changes using IAM Access Analyser to check for unintended access.

Update Systems

- Implement stricter CloudTrail event filtering for IAM activity.
- Update AWS Config rules to trigger auto-remediation for specific policy violations.

6. LESSONS LEARNT

Post-Incident Review

- Identify gaps in detection and response processes.
- Determine if privilege misuse, credential theft or insider threats caused the change.

Enhance Defences

- Automate policy review with custom Lambda functions triggered by IAM changes.
- Increase training for administrators on secure IAM practices.

AWS PLAYBOOK: RESPONDING TO VPC NETWORK TRAFFIC ANOMALIES

1. PREPARATION

Baseline Security Measures

- **VPC Flow Logs**:
  - Enable Flow Logs for all VPCs and route them to a secure S3 bucket or CloudWatch.
  - Ensure logs capture traffic at the subnet, instance and ENI levels.
- **AWS GuardDuty**:
  - Enable GuardDuty to monitor for anomalous network activity, such as outbound traffic to suspicious IPs or regions.
- **Network ACLs and Security Groups**:
  - Enforce least-privilege rules for inbound and outbound traffic.
  - Regularly audit rules to avoid overly permissive configurations (e.g., 0.0.0.0/0).

Monitoring and Alerts

- **Set up Amazon CloudWatch Alarms to:**
  - Detect sudden spikes in network traffic or unexpected data egress.
  - Monitor unusual connections to uncommon ports or destinations.
- Use Amazon Inspector to identify misconfigurations or vulnerabilities in resources exposed to the internet.

Auditing and Simulation

- Regularly test the network security posture with simulated anomalies:
  - Unauthorised connections to sensitive subnets.
  - Traffic patterns mimicking exfiltration or lateral movement.

2. DETECT

Indicators of VPC Network Traffic Anomalies

- **Flow Log Patterns**:
  - Unexpected traffic spikes from specific instances or subnets.
  - High volumes of outbound traffic to suspicious regions or IPs.
  - Traffic to unusual ports or protocols not typically used by the application.
- **GuardDuty Findings**:
  - Alerts for data exfiltration, port scanning or anomalous behaviour.
- **CloudWatch Metrics**:
  - Alarms for high network throughput or packet drops in specific subnets.

Employee or Partner Notifications

- Reports of degraded application performance due to potential DDoS or unauthorised traffic.

3. ANALYSE

Review Logs and Alerts

- **VPC Flow Logs**:
  - Identify anomalous traffic patterns, including source and destination IPs, ports and protocols.
- **GuardDuty Findings**:
  - Analyse alerts for malicious IPs, unusual data transfer patterns or brute force attempts.
- **CloudTrail**:
  - Verify if recent configuration changes in the VPC, security groups or routes correlate with the anomaly.

Impact Assessment

- Determine if the anomaly indicates:
  - Data exfiltration to unauthorised locations.
  - Lateral movement within the VPC.
  - DDoS activity impacting application performance.

4. CONTAIN / ERADICATE

Containment Steps

- **Quarantine Compromised Resources**:
  - Isolate affected instances by applying restrictive security group rules.
- **Block Malicious Traffic**:
  - Update Network ACLs or security group rules to deny traffic from malicious IPs.
  - Use **AWS WAF** to block patterns of unauthorised requests.

Eradication Steps

- Terminate compromised instances after capturing forensic data (e.g., memory snapshots, disk images).
- Validate route tables and security group rules to ensure no unauthorised changes persist.

5. RECOVER

Restore Secure State

- Launch new instances from trusted AMIs if compromised instances were terminated.
- Reapply baseline security configurations to ensure all rules align with the least-privilege principle.

Update Systems

- Configure automated remediation using **AWS Systems Manager** or Lambda to respond to similar anomalies in the future.
- Refine monitoring rules in GuardDuty and CloudWatch.

6. LESSONS LEARNT

Post-Incident Review

- Identify how the anomaly was introduced:
    - Was it a result of configuration drift, compromised credentials or malicious intent?
- Evaluate the response timeline:
    - How quickly was the anomaly detected, contained and resolved?

Enhance Defences

- Strengthen anomaly detection rules by using machine learning tools like **Amazon Lookout for Metrics**.
- Conduct employee training to improve awareness of network security practices.

AWS PLAYBOOK: RESPONDING TO UNUSUAL RDS DATABASE CONFIGURATION CHANGES

1. PREPARATION

Baseline Security Configurations

- **Enable GuardDuty:**
  - Activate across all accounts and regions to detect anomalous RDS-related activities (e.g., unusual API calls).
- **IAM Policies:**
  - Enforce least privilege principles for database configuration management.
  - Restrict permissions to modify RDS configurations to a specific group of administrators.
  - Implement Service Control Policies (SCPs) to limit access to sensitive RDS configurations.
- **CloudTrail Logging:**
  - Enable AWS CloudTrail for all regions to capture ModifyDBInstance and related events.
  - Store logs in an encrypted, private S3 bucket with strict access controls for auditing purposes.

Monitoring and Alerts

- Use **Amazon CloudWatch** to:
  - Set alarms for changes to RDS configurations, such as:
    - Modifications to DB security groups.
    - Changes to DB instance classes or engine versions.
    - Changes in backup retention periods or parameter groups.
  - Track unexpected connections from unusual IPs or geolocations.

Tagging Standards

- Apply consistent tags (e.g., Owner, Environment, Compliance) for all RDS instances to easily identify legitimate configurations.

Training and Awareness

- Train database administrators and operations teams to:
  - Recognise unauthorised configuration changes.
  - Follow predefined incident response procedures.
- Conduct regular simulations involving unusual RDS configuration scenarios.

2. DETECT

Anomalous Configuration Change Indicators

- **GuardDuty Findings:**
  - "Unusual access pattern detected for RDS instance."
  - "Potential credential compromise related to RDS API calls."
- **CloudTrail Logs:**
  - Review ModifyDBInstance, RebootDBInstance or DeleteDBInstance events for:
    - Unusual IAM users, IP addresses or geolocations initiating the changes.
    - Modifications outside of standard maintenance windows.
- **CloudWatch Metrics:**
  - Look for unexpected changes in:
    - CPU or memory utilisation after configuration updates.
    - Connection spikes that might indicate unauthorised access.

Potential Triggers

- Configuration updates to:
  - Enable public accessibility unexpectedly.
  - Modify security group rules allowing broader access.
  - Alter database encryption settings (e.g., turning off encryption).
- Unusual Costs:
  - Significant billing increases due to upscaling or misuse.

3. ANALYSE

Review Logs

- **CloudTrail:**
  - Identify who initiated the configuration changes and their IP address.
  - Look for permission changes or credential usage anomalies before the event.
- **RDS Audit Logs (if enabled):**
  - Check for unusual SQL queries or connections related to the incident.
- **VPC Flow Logs:**
  - Review outbound and inbound traffic patterns for RDS instances.

Categorise the Incident

- Determine the root cause:
  - Credential compromise?
  - Misconfigured IAM policies?
  - Malicious insiders?

Assess Impact

- Evaluate:
    - If sensitive data was exposed or exfiltrated.
    - Potential unauthorised access to database content.

4. CONTAIN / ERADICATE

Containment Steps

- **Restrict access:**
    - Modify security group rules to block unauthorised IPs or regions.
    - Temporarily disable public access if enabled.
- **Lock down accounts:**
    - Revoke credentials or access tokens used for the unauthorised configuration changes.
- **Quarantine the database:**
    - Move the RDS instance to a dedicated security group for investigation.

Eradication Steps

- Roll back unauthorised configuration changes using backups or parameter group snapshots.
- Rotate all affected IAM credentials and database access credentials.
- Scan the account for additional unauthorised activity.

5. RECOVER

Validate Recovery

- Ensure all database configurations are consistent with the baseline standards.
- Verify that no unauthorised RDS instances or configurations persist.

Restore Configurations

- Reapply compliance configurations such as:
    - Tagging standards.
    - Security group rules.
    - Encryption settings.
- Update CloudTrail filters and alarms for better detection of similar configuration changes.

6. LESSONS LEARNT

Post-Incident Review

- Identify the root cause:
    - Weak IAM controls?
    - Gaps in monitoring or alerting?
- Update the playbook to address any identified shortcomings.

Enhanced Security Practices

- Automate detection of unauthorised changes with AWS Lambda or Config Rules.
- Increase the frequency of RDS audits and IAM policy reviews.
- Implement anomaly detection for configuration changes using AWS Config or third-party tools.

AWS PLAYBOOK: RESPONDING TO UNEXPECTED CHANGES IN CLOUDFORMATION STACKS

1. PREPARATION

Baseline Security Configurations

- **Enable GuardDuty:**
  - Enable across all accounts and regions to detect unusual CloudFormation API calls (e.g., UpdateStack or DeleteStack).
- **IAM Policies:**
  - Restrict permissions for stack modifications (e.g., UpdateStack, DeleteStack) to specific roles or users.
  - Use Service Control Policies (SCPs) to block sensitive changes, such as creating stacks in unauthorised regions.
- **CloudTrail Logging:**
  - Ensure AWS CloudTrail is enabled for all regions to capture CloudFormation events.
  - Store logs in a private, encrypted S3 bucket with strict access controls for audit purposes.

Monitoring and Alerts

- Use **Amazon CloudWatch** to:
  - Set alarms for:
    - Unexpected UpdateStack or DeleteStack events.
    - Stack creations in unapproved regions.
  - Monitor CloudFormation stack drift using AWS Config rules.

Tagging Standards

- Enforce consistent tagging for stacks (e.g., Owner, Environment, Purpose) to easily identify legitimate stack changes.

Training and Awareness

- Train teams to:
  - Understand stack drift and its implications.
  - Respond to unauthorised stack changes using this playbook.
- Conduct regular simulation exercises for CloudFormation-based incidents.

2. DETECT

Anomalous Change Indicators

- **GuardDuty Findings:**
  - "Unusual API calls detected for CloudFormation."
  - "CloudFormation stack changes from an unapproved region or IP address."
- **CloudTrail Logs:**
  - Review UpdateStack, DeleteStack or CreateStack events for:
    - Unknown users or roles initiating changes.
    - API calls from unusual IP addresses or geolocations.
- **CloudWatch Metrics:**
  - Detect anomalies such as:
    - Stacks being modified outside approved change windows.
    - Unexpected changes to critical resources (e.g., IAM roles, VPC configurations).

Potential Triggers

- **Stack Drift:**
  - Unintended changes detected in stack-managed resources.
- **Unusual Stack Activity:**
  - Unfamiliar users or services creating or modifying stacks.
  - Unexpected stack deletions or resource removals.
- **Cost Anomalies:**
  - Sudden increases in billing due to unauthorised stack creation or scaling.

3. ANALYSE

Review Logs

- **CloudTrail:**
  - Identify who initiated the stack changes and their IP address.
  - Check for prior actions by the same user or role, such as credential or policy changes.
- **Stack Change Sets:**
  - Review the change set for:
    - Resources affected.
    - Alterations to IAM roles, security groups or networking configurations.
- **CloudFormation Drift Detection:**
  - Identify resources that have deviated from the stack's intended state.

Categorise the Incident

- Determine if the incident was caused by:
  - Credential compromise?
  - Malicious insider activity?
  - Poorly configured IAM policies?

Assess Impact

- Evaluate:
    - The scope of resource changes (e.g., IAM, VPC or EC2).
    - Potential exposure or disruption caused by the incident.

4. CONTAIN / ERADICATE

Containment Steps

- **Restrict Access:**
    - Temporarily revoke permissions for users or roles involved in the incident.
- **Rollback Changes:**
    - If possible, use the CloudFormation rollback feature to restore the stack to its previous state.
- **Isolate Impacted Resources:**
    - Apply restrictive security group rules or disable affected services.

Eradication Steps

- **Remove Unauthorised Stacks:**
    - Delete unauthorised stacks or resources created by the changes.
- **Update IAM Policies:**
    - Adjust IAM roles or SCPs to prevent future unauthorised actions.
- **Investigate Drift:**
    - Correct resource configurations that differ from the expected stack state.

5. RECOVER

Validate Recovery

- Confirm that all affected stacks are aligned with their original templates and configurations.
- Ensure no unauthorised resources remain active in the environment.

Restore Configurations

- Reinforce tagging standards for all stacks.
- Update CloudTrail filters to monitor stack changes more effectively.

6. LESSONS LEARNT

Post-Incident Review

- Analyse the root cause:
    - Was it a credential compromise or a policy misconfiguration?
    - Were there gaps in monitoring or alerting?
- Update the playbook to include:
    - Improved detection rules.
    - Enhanced response processes.

Enhanced Security Practices

- Automate drift detection and remediation using AWS Config or third-party tools.
- Schedule regular audits of IAM policies and CloudFormation configurations.
- Increase the frequency of security training for teams managing CloudFormation.

AWS PLAYBOOK: RESPONDING TO UNAUTHORISED ACCESS TO SECRETS MANAGER

1. PREPARATION

Baseline Security Configurations

- **IAM Policies:**
  - Apply the principle of least privilege for accessing Secrets Manager.
  - Use specific resource-based policies to restrict secret access to approved users or roles.
  - Implement AWS Identity Center (formerly AWS SSO) for centralised access control.
- **Encryption:**
  - Ensure secrets are encrypted using AWS Key Management Service (KMS) with customer-managed keys (CMKs).
  - Restrict KMS key usage to specific roles or services.
- **CloudTrail Logging:**
  - Enable AWS CloudTrail for all regions to log GetSecretValue, CreateSecret and UpdateSecret API calls.
  - Store logs in a secure, encrypted S3 bucket with limited access.
- **Secrets Rotation:**
  - Automate secret rotation using AWS Secrets Manager.
  - Ensure rotation frequency aligns with organisational policies and industry best practices.

Monitoring and Alerts

- **Amazon CloudWatch:**
  - Set alarms for:
    - Unusual GetSecretValue calls.
    - API activity from unknown users or regions.
  - Enable anomaly detection on Secrets Manager API call metrics.
- **AWS Config:**
  - Create rules to detect public exposure of secrets or weak resource-based policies.

Tagging Standards

- Tag secrets with metadata (e.g., Owner, Environment, Purpose) to simplify identification during incidents.

Training and Awareness

- Educate administrators on:
    - Secure use of Secrets Manager.
    - Recognising unauthorised activity and responding to incidents.
- Conduct simulations for unauthorised access scenarios.

2. DETECT

Anomalous Access Indicators

- **CloudTrail Logs:**
    - Unusual GetSecretValue or ListSecrets calls:
        - From unknown users or IPs.
        - Outside approved regions or hours.
- **GuardDuty Findings:**
    - "Unauthorised API calls detected in Secrets Manager."
    - "IAM principal attempting secrets access from a known malicious IP address."
- **CloudWatch Metrics:**
    - Spikes in API calls related to Secrets Manager.
    - Access attempts using unapproved IAM roles or temporary credentials.

Potential Triggers

- **Compromised IAM Credentials:**
    - Unauthorised access due to credential theft or misuse.
- **Policy Misconfiguration:**
    - Overly permissive IAM or resource policies allowing unauthorised access.
- **Malicious Insider Activity:**
    - Intentional misuse of valid access credentials.

3. ANALYSE

Review Logs

- **CloudTrail:**
    - Identify who accessed the secret, their IP address and the time of access.
    - Check if additional IAM actions (e.g., privilege escalation) preceded the incident.
- **Secrets Manager Activity:**
    - Analyse GetSecretValue, UpdateSecret and DeleteSecret API calls for suspicious activity.
- **VPC Flow Logs:**
    - Look for outbound traffic from resources accessing Secrets Manager to unknown IPs or regions.

Categorise the Incident

- Determine the nature of unauthorised access:
  - Credential compromise?
  - Misconfigured policies?
  - Malicious insider activity?

Assess Impact

- Evaluate:
  - Secrets accessed and their criticality (e.g., database credentials, API keys).
  - Potential data exfiltration or compromise of downstream systems.

4. CONTAIN / ERADICATE

Containment Steps

- **Revoke Access:**
  - Immediately disable the IAM user or role involved in unauthorised access.
  - Revoke any active sessions for compromised credentials.
- **Rotate Secrets:**
  - Rotate all affected secrets using AWS Secrets Manager's rotation feature.
  - Notify impacted teams of updated secret values.
- **Update Policies:**
  - Restrict access to Secrets Manager with tighter IAM or resource-based policies.
  - Implement conditions in policies (e.g., IP restrictions, MFA requirements).

Eradication Steps

- **Audit IAM Policies:**
  - Identify and correct overly permissive roles or policies.
  - Enable AWS Identity Center (SSO) for enhanced identity governance.
- **Scan for Additional Threats:**
  - Use GuardDuty or third-party tools to detect ongoing malicious activity.
  - Investigate other AWS services for compromise using the accessed secrets.

5. RECOVER

Validate Recovery

- Confirm no unauthorised access to secrets or dependent services remains active.
- Ensure all secrets are rotated and downstream systems are reconfigured with new credentials.

Restore Configurations

- Harden IAM policies and enforce role-based access controls.
- Enable resource tagging for easier secret management and monitoring.

6. LESSONS LEARNT

Post-Incident Review

- Identify root causes:
  - Credential compromise?
  - Misconfiguration?
- Analyse response gaps:
  - Was unauthorised access detected and mitigated promptly?
  - Were alerts configured effectively?

Enhanced Security Practices

- Automate detection of unusual API activity using AWS Lambda or third-party tools.
- Perform regular IAM policy reviews and secrets rotation audits.
- Increase monitoring for secrets accessed during the incident.

AWS PLAYBOOK: RESPONDING TO UNAUTHORISED S3 OBJECT DELETION OR MODIFICATION

1. PREPARATION

Baseline Security Configurations

- **S3 Bucket Policies and Permissions:**
  - Apply the principle of least privilege to S3 buckets.
  - Deny public access to buckets unless explicitly required.
  - Enable bucket versioning to retain previous versions of objects.
  - Use multi-factor authentication (MFA) for sensitive object deletions.
- **Encryption:**
  - Encrypt objects at rest using S3-managed keys or AWS Key Management Service (KMS).
  - Enforce TLS encryption for data in transit.
- **CloudTrail Logging:**
  - Enable AWS CloudTrail for all regions to log S3-related API calls, such as DeleteObject and PutObject.
  - Store CloudTrail logs in a secure, immutable S3 bucket.
- **S3 Access Logs:**
  - Enable server access logging for all S3 buckets to capture access requests and usage patterns.

Monitoring and Alerts

- **Amazon CloudWatch Alarms:**
  - Set alarms for:
    - Unusual spikes in DeleteObject or PutObject API calls.
    - API activity from unknown users, regions or IPs.
  - Monitor metrics for data transfer and object storage changes.
- **GuardDuty:**
  - Enable GuardDuty to detect:
    - “S3 bucket policy change” findings.
    - “Unusual data access patterns.”

Tagging Standards

- Use consistent tagging (e.g., Owner, Environment, Data Classification) to identify and categorise buckets and objects easily.

Training and Awareness

- Train administrators to recognise unauthorised S3 activity and respond effectively.

- Conduct regular simulations involving object deletions or modifications.

2. DETECT

Indicators of Unauthorised Deletion or Modification

- **CloudTrail Logs:**
  - Review API calls like DeleteObject, PutObject or DeleteBucket.
  - Look for calls from unusual IP addresses or regions.
- **S3 Access Logs:**
  - Detect unexpected object deletions, modifications or data transfer.
- **GuardDuty Findings:**
  - Alerts for unusual access behavior or policy changes.
- **CloudWatch Metrics:**
  - Sudden changes in bucket sise or object count.

Potential Triggers

- **Compromised IAM Credentials:**
  - Unauthorised access due to stolen or misused credentials.
- **Misconfigured Bucket Policies:**
  - Overly permissive policies enabling unintended access.
- **Malicious Insider Activity:**
  - Deliberate deletions or modifications by authorised users.

3. ANALYSE

Review Logs

- **CloudTrail:**
  - Identify the user or role that performed the deletion/modification.
  - Check the source IP, timestamp and region of the API calls.
- **S3 Access Logs:**
  - Pinpoint affected objects and the method of access.
- **VPC Flow Logs:**
  - Analyse unusual outbound traffic patterns or data exfiltration attempts.

Categorise the Incident

- Determine whether the action was due to:
  - Credential compromise.
  - Misconfigured bucket policies.
  - Malicious insider activity.

Assess Impact

- Evaluate the scope of deleted or modified objects:
    - Criticality of the data.
    - Dependencies on the data by other systems or services.

4. CONTAIN / ERADICATE

Containment Steps

- **Restrict Access:**
    - Revoke IAM credentials associated with the unauthorised action.
    - Temporarily block access to the affected bucket using bucket policies.
- **Enable Bucket Versioning:**
    - Recover deleted or overwritten objects using versioned backups.
- **Review and Update Policies:**
    - Restrict bucket policies to allow access only from known, trusted accounts or IP ranges.
    - Implement conditions requiring MFA for sensitive actions.

Eradication Steps

- **Audit IAM Roles and Policies:**
    - Identify and remove overly permissive roles or policies.
- **Scan for Additional Activity:**
    - Use GuardDuty and CloudTrail logs to detect any further unauthorised actions.
- **Update Key Management:**
    - Rotate access keys and KMS keys used for encryption.

5. RECOVER

Validate Recovery

- Confirm that all affected objects are restored using versioned backups or other recovery methods.
- Ensure no unauthorised access persists to the bucket or its objects.

Restore Configurations

- Reinforce bucket policies, IAM roles and MFA for sensitive actions.
- Ensure CloudTrail and S3 Access Logs are capturing all relevant activity.

6. LESSONS LEARNT

Post-Incident Review

- Identify root causes:
  - Credential compromise?
  - Misconfigured permissions?
- Assess response effectiveness:
  - Were unauthorised actions detected promptly?
  - Were response steps adequate to mitigate the impact?

Enhanced Security Practices

- Automate detection of unauthorised S3 actions using AWS Lambda or EventBridge rules.
- Regularly review and audit S3 bucket policies and permissions.
- Increase training frequency for administrators on securing S3 resources.

AWS PLAYBOOK: RESPONDING TO LAMBDA EXECUTION ANOMALIES

1. PREPARATION

Baseline Security Configurations

- **IAM Roles and Permissions:**
  - Apply the principle of least privilege for Lambda execution roles.
  - Restrict permissions to invoke Lambda functions to trusted entities only.
- **Environment Variables:**
  - Encrypt sensitive environment variables using AWS KMS.
  - Limit access to decrypt KMS-encrypted variables to trusted roles.
- **Network Configuration:**
  - Restrict Lambda access to private VPC subnets when interacting with internal resources.
  - Use security groups to limit outbound traffic.
- **Logging and Monitoring:**
  - Enable detailed Lambda function logs in Amazon CloudWatch.
  - Use AWS X-Ray for tracing Lambda function performance and dependencies.

Monitoring and Alerts

- **Amazon CloudWatch Alarms:**
  - Monitor anomalies like:
    - Spikes in Lambda invocation rates.
    - Increased execution errors or durations.
    - High concurrency levels.
- **GuardDuty Findings:**
  - Detect suspicious API calls to Lambda, such as unauthorised invocations or modifications.
- **AWS Config Rules:**
  - Validate that Lambda functions comply with security best practices, e.g., using encrypted environment variables.

Training and Awareness

- Train teams on secure Lambda development practices and anomaly detection.
- Conduct drills simulating Lambda abuse scenarios, such as privilege escalation or malicious payloads.

2. DETECT

Anomalous Execution Indicators

- **CloudWatch Logs:**
  - Unusual error messages or exceptions.
  - Unexpected invocation patterns (e.g., unusually frequent or rare).
- **CloudWatch Metrics:**
  - Sudden spikes in:
    - Invocation count.
    - Execution duration or memory usage.
    - Errors or throttles.
- **GuardDuty Findings:**
  - Alerts for:
    - "Unauthorised Lambda invocation."
    - "Unusual access patterns or data exfiltration attempts."

Potential Triggers

- **Compromised IAM Role:**
  - Malicious actors invoking the function or altering its configuration.
- **Malicious Payloads:**
  - Functions performing unauthorised actions or exfiltrating data.
- **Code Vulnerabilities:**
  - Exploitation of insecure code leading to privilege escalation or misuse.

3. ANALYSE

Review Logs

- **CloudTrail Logs:**
  - Examine InvokeFunction and UpdateFunctionConfiguration events for unauthorised users or IPs.
- **CloudWatch Logs:**
  - Look for anomalous output or execution patterns.
- **AWS X-Ray:**
  - Trace execution flows to identify unexpected dependencies or calls.

Categorise the Incident

- Determine if the anomaly was caused by:
  - A compromised IAM role or user.
  - Malicious or unauthorised invocation.
  - Exploitation of code vulnerabilities.

Assess Impact

- Investigate whether:

    - Unauthorised data access or exfiltration occurred.
    - The function was used to launch further attacks or privilege escalation.

4. CONTAIN / ERADICATE

Containment Steps

- **Restrict Access:**
    - Disable the affected Lambda function by setting its concurrency to zero.
    - Update IAM policies to deny access to the compromised function.
- **Isolate Resources:**
    - Remove the function from associated event triggers (e.g., S3, DynamoDB).
    - Revoke any affected IAM credentials.
- **Stop Malicious Activity:**
    - Monitor VPC Flow Logs to detect and block suspicious traffic.
    - Update security groups and route tables to prevent data exfiltration.

Eradication Steps

- **Inspect and Update Code:**
    - Review function code for vulnerabilities or malicious payloads.
    - Deploy patched versions of the function with secure configurations.
- **Rotate Credentials:**
    - Rotate access keys and secrets associated with the Lambda function.
- **Audit Resources:**
    - Scan the AWS account for additional compromised resources or configurations.

5. RECOVER

Validate Recovery

- Test the updated Lambda function to confirm secure and expected behavior.
- Confirm no unauthorised invocations or anomalous metrics are observed.

Restore Configurations

- Reinforce permissions for Lambda execution roles and associated triggers.
- Update monitoring rules to detect similar anomalies earlier.

6. LESSONS LEARNT

Post-Incident Review

- Root Cause Analysis:
    - Identify the origin of the anomaly (e.g., compromised credentials, insecure code).
- Response Effectiveness:
    - Evaluate whether detection and response steps were adequate and timely.

Enhanced Security Practices

- Automate anomaly detection using AWS Lambda or EventBridge rules.
- Enforce regular security reviews for Lambda code and configurations.
- Increase training frequency for developers and administrators on secure Lambda practices.

AWS PLAYBOOK: RESPONDING TO ROGUE INSTANCES

1. PREPARATION

Baseline Security Configurations

- **IAM Policies:**
  - Implement least privilege for instance launch and management permissions.
  - Restrict EC2 launch permissions using IAM roles and service control policies (SCPs).
  - Deny access to regions not actively in use.
- **GuardDuty:**
  - Enable GuardDuty across all accounts and regions to detect anomalous instance activity.
- **CloudTrail Logging:**
  - Ensure AWS CloudTrail is enabled for all regions to log RunInstances events.
  - Store logs in an encrypted S3 bucket with strict access controls.
- **Tagging Standards:**
  - Enforce mandatory tagging for all EC2 instances (e.g., Owner, Environment).
  - Use AWS Config to flag untagged or incorrectly tagged instances.
- **VPC Security:**
  - Restrict EC2 internet access using security groups and NAT gateways.
  - Enable VPC Flow Logs to monitor instance traffic.

Monitoring and Alerts

- **CloudWatch Alarms:**
  - Set alarms for:
    - Instances launched in unapproved regions.
    - Unusual spikes in EC2 usage (e.g., CPU, network or disk activity).
- **GuardDuty Findings:**
  - Alert for behaviors like data exfiltration or botnet activity.

Training and Awareness

- Train administrators to recognise and report suspicious EC2 activity.
- Conduct regular simulations of rogue instance scenarios to refine response processes.

2. DETECT

Anomalous Instance Indicators

- **CloudTrail Logs:**

    - Look for unauthorised RunInstances events:
        - Unexpected IAM users or roles initiating launches.
        - Use of unfamiliar AMIs, key pairs or security groups.
- **GuardDuty Findings:**
    - Alerts for:
        - "EC2 instance behaving as a botnet controller."
        - "Unusual geolocations or ports in network traffic."
- **CloudWatch Metrics:**
    - Spikes in:
        - EC2 usage.
        - Unusual traffic patterns in VPC Flow Logs.
- **Billing Dashboard:**
    - Sudden cost increases from unapproved instance launches.

Potential Triggers

- **Credential Compromise:**
    - Attackers leveraging stolen credentials to launch instances.
- **Misconfigured IAM Policies:**
    - Overly permissive policies allowing unauthorised launches.
- **Insider Threats:**
    - Malicious or accidental instance launches by internal users.

3. ANALYSE

Review Logs

- **CloudTrail:**
    - Determine who initiated the rogue instance.
    - Identify IP addresses, regions and user agents associated with the activity.
- **VPC Flow Logs:**
    - Monitor the rogue instance for suspicious traffic patterns.
- **Instance Metadata:**
    - Review details such as AMI ID, instance type and attached security groups.

Categorise the Incident

- Classify whether the rogue instance was caused by:
    - A credential compromise.
    - IAM policy misconfigurations.
    - Malicious insiders or external attackers.

Assess Impact

- Check for:
    - Unauthorised access to sensitive resources.
    - Lateral movement within the network.
    - Data exfiltration or communication with known malicious IPs.

4. CONTAIN / ERADICATE

Containment Steps

- **Isolate the Instance:**
    - Move the rogue instance to a quarantine VPC or assign a restrictive security group.
    - Disable internet access by updating route tables or NAT gateways.
- **Restrict Permissions:**
    - Immediately revoke IAM credentials or access keys used to launch the instance.
    - Implement a temporary deny-all SCP for the affected account or region.

Eradication Steps

- **Terminate the Rogue Instance:**
    - Confirm the instance is no longer communicating with sensitive resources before termination.
- **Audit Permissions:**
    - Identify and update IAM policies that allowed unauthorised launches.
- **Scan the Account:**
    - Review other instances and services for signs of compromise.

5. RECOVER

Validate Recovery

- Ensure no rogue instances remain in the environment.
- Monitor for any further unauthorised RunInstances events in CloudTrail.

Restore Configurations

- Enforce tagging standards and restrict regions for instance launches.
- Update monitoring rules to detect similar incidents earlier.
- Rotate IAM credentials if compromise was confirmed.

6. LESSONS LEARNT

Post-Incident Review

- **Root Cause Analysis:**
    - Assess how the rogue instance was launched (e.g., credential compromise or policy misconfiguration).
- **Process Evaluation:**
    - Review the effectiveness of detection and response measures.

Enhanced Security Practices

- Automate detection of unauthorised launches using AWS Lambda and EventBridge.
- Conduct regular audits of IAM policies and credentials.
- Increase monitoring for unusual usage patterns in CloudTrail and billing dashboards.

AWS PLAYBOOK: RESPONDING TO UNAUTHORISED SECURITY GROUP CHANGES

1. PREPARATION

Baseline Security Configurations

- **IAM Policies:**
  - Enforce least privilege by restricting who can modify security groups.
  - Implement SCPs to limit actions on security groups to authorised users and services.
- **GuardDuty:**
  - Enable GuardDuty to detect anomalous activity related to security groups.
- **CloudTrail Logging:**
  - Ensure CloudTrail is enabled for all regions to track AuthoriseSecurityGroupIngress, AuthoriseSecurityGroupEgress and RevokeSecurityGroupIngress/Egress events.
  - Configure logs to be encrypted and stored in a private S3 bucket.
- **AWS Config:**
  - Set up AWS Config rules to check for:
    - Open or overly permissive security group rules.
    - Unauthorised IP ranges or protocols.
- **VPC Security:**
  - Use network ACLs as an additional layer of protection to restrict traffic.

Monitoring and Alerts

- **CloudWatch Alarms:**
  - Create alarms for changes to security groups with:
    - Wide-open ingress/egress rules (e.g., 0.0.0.0/0 on all ports).
    - High-risk ports exposed (e.g., SSH on 0.0.0.0/0).
- **AWS Config Rules Alerts:**
  - Monitor non-compliant security groups and unauthorised changes.

Training and Awareness

- Train administrators on the importance of securing security groups.
- Conduct regular drills to simulate scenarios involving unauthorised security group changes.

2. DETECT

Indicators of Unauthorised Changes

- **CloudTrail Logs:**
  - Look for unauthorised API calls such as:
    - AuthoriseSecurityGroupIngress/Egress.
    - RevokeSecurityGroupIngress/Egress.
  - Check for unusual IP addresses, regions or IAM roles initiating the changes.
- **AWS Config Non-Compliance:**
  - Detect violations of security group compliance rules (e.g., open ports or IP ranges).
- **GuardDuty Findings:**
  - Alerts for:
    - "UnauthorisedAccess: EC2/NetworkPermissions".
    - Anomalous traffic patterns caused by newly opened ports.
- **CloudWatch Metrics:**
  - Monitor for spikes in network traffic following security group changes.

Potential Triggers

- Overly permissive changes to allow unrestricted access.
- Addition of unfamiliar IP ranges or protocols.
- Removal of critical security group rules, reducing protection.

3. ANALYSE

Review Logs

- **CloudTrail Logs:**
  - Identify who initiated the security group change.
  - Determine the API call details, including the time, IP address and user agent.
- **AWS Config Compliance History:**
  - Review historical compliance data to identify when and how rules were violated.
- **VPC Flow Logs:**
  - Check for new or unusual traffic patterns resulting from the modified rules.

Categorise the Incident

- Determine whether the change was:
  - Accidental.
  - Due to credential compromise.
  - The result of a malicious insider or external attacker.

Assess Impact

- Identify services or instances affected by the change.

- Check for potential unauthorised access or data exfiltration.

4. CONTAIN / ERADICATE

Containment Steps

- **Revert Security Group Changes:**
  - Immediately revoke unauthorised rules and restore the original configuration.
- **Restrict Access:**
  - Temporarily block IAM users or roles that made the change.
  - Apply restrictive security group rules until the incident is resolved.

Eradication Steps

- **Audit IAM Permissions:**
  - Identify and fix overly permissive IAM roles or policies.
- **Review Affected Accounts:**
  - Scan for additional unauthorised activity in the affected account.

5. RECOVER

Validate Recovery

- Ensure that all security groups comply with organisational policies.
- Monitor for further unauthorised changes in CloudTrail and AWS Config.

Restore Configurations

- Update and enforce security group baselines.
- Apply automated tools to revert unauthorised changes instantly (e.g., AWS Lambda or Config Remediation Actions).

6. LESSONS LEARNT

Post-Incident Review

- Conduct a root cause analysis to determine how unauthorised changes were made.
- Identify gaps in monitoring, logging or IAM permissions.

Enhanced Security Practices

- Automate detection and response with Lambda to roll back unauthorised changes.
- Increase the frequency of IAM audits to prevent privilege escalation.

- Regularly test security configurations and update training for administrators.

AWS PLAYBOOK: RESPONDING TO UNAUTHORISED EBS VOLUME SNAPSHOTS

1. PREPARATION

Baseline Security Configurations

- **IAM Policies:**
  - Restrict who can create, share or delete EBS snapshots using least-privilege IAM policies.
  - Use Service Control Policies (SCPs) to enforce account-wide restrictions on EBS snapshot operations.
- **Encryption:**
  - Enforce encryption of EBS snapshots using AWS KMS keys.
  - Restrict access to KMS keys to authorised users and services only.
- **CloudTrail Logging:**
  - Enable CloudTrail to log CreateSnapshot, DeleteSnapshot, CopySnapshot and ModifySnapshotAttribute events.
  - Store CloudTrail logs in a secure, encrypted S3 bucket for analysis.
- **AWS Config:**
  - Set rules to ensure that:
    - All EBS snapshots are encrypted.
    - Public sharing of snapshots is prohibited.

Monitoring and Alerts

- **CloudWatch Alarms:**
  - Set alarms for:
    - Creation of unencrypted snapshots.
    - Snapshots shared publicly or with external accounts.
- **GuardDuty:**
  - Enable GuardDuty to detect anomalies related to unauthorised snapshot operations or data exfiltration.

Tagging Standards

- Enforce consistent tagging for snapshots (e.g., Owner, Purpose, Environment) to identify legitimate snapshots easily.

Training and Awareness

- Train administrators to recognise suspicious EBS snapshot activity.
- Conduct drills simulating scenarios involving unauthorised snapshot creation or sharing.

2. DETECT

Indicators of Unauthorised Snapshots

- **CloudTrail Logs:**
  - Look for suspicious API calls, including:
    - CreateSnapshot.
    - ModifySnapshotAttribute (e.g., making snapshots public).
  - Unusual IP addresses, regions or IAM roles involved in snapshot operations.
- **AWS Config Compliance:**
  - Detect non-compliant snapshots, such as unencrypted or publicly shared ones.
- **GuardDuty Findings:**
  - Alerts for anomalies like unauthorised data access or data exfiltration.

Potential Triggers

- Snapshots created by unauthorised users or roles.
- Snapshots shared publicly or copied to external accounts.
- Unencrypted snapshots appearing in the environment.

3. ANALYSE

Review Logs

- **CloudTrail Logs:**
  - Determine who created or modified the snapshot.
  - Review the API call details, including time, IP address and user agent.
- **AWS Config Compliance History:**
  - Check historical data for compliance violations involving snapshots.
- **Snapshot Metadata:**
  - Review snapshot details, including encryption status, sharing permissions and tags.

Categorise the Incident

- Determine if the activity was:
  - Accidental (e.g., human error).
  - Due to credential compromise.
  - The result of malicious insider or external attacker actions.

Assess Impact

- Identify volumes or data associated with the snapshot.

- Check if sensitive data is exposed or if snapshots were shared externally.

4. CONTAIN / ERADICATE

Containment Steps

- **Restrict Access:**
  - Immediately revoke permissions for IAM users or roles involved in the incident.
- **Remove Public Access:**
  - Modify snapshot attributes to remove public or unauthorised sharing.

Eradication Steps

- **Delete Unauthorised Snapshots:**
  - Terminate any snapshots created without authorisation.
- **Audit IAM Permissions:**
  - Identify and resolve overly permissive roles or policies.
- **Scan for Further Unauthorised Activity:**
  - Check for additional snapshots or related anomalies.

5. RECOVER

Validate Recovery

- Confirm that no unauthorised snapshots or permissions remain.
- Ensure compliance with tagging and encryption policies.

Restore Configurations

- Enforce automatic remediation using AWS Config for public or unencrypted snapshots.
- Update IAM policies to restrict snapshot operations more effectively.

6. LESSONS LEARNT

Post-Incident Review

- Identify gaps in monitoring, logging or IAM policy enforcement.
- Review the root cause, such as weak access control or credential compromise.

Enhanced Security Practices

- Automate snapshot management with tools like AWS Lambda to detect and remediate unauthorised actions.
- Increase the frequency of IAM role audits and credential rotation.
- Educate users on the risks of improper snapshot management.

AWS PLAYBOOK: RESPONDING TO SUSPICIOUS LOGIN ACTIVITY

1. PREPARATION

Baseline Security Configurations

- **MFA Enforcement:**
  - Require Multi-Factor Authentication (MFA) for all AWS accounts, especially for privileged IAM users and roles.
- **IAM Policies:**
  - Use least-privilege access for IAM users and roles.
  - Implement Service Control Policies (SCPs) to restrict access to sensitive regions and actions.
- **Password Policies:**
  - Enforce strong password policies, including length, complexity and rotation.
- **CloudTrail Logging:**
  - Enable CloudTrail to log all management console sign-in events across all regions.
  - Store logs in an encrypted S3 bucket with access restricted to administrators.
- **AWS Config:**
  - Set rules to check for security best practices, such as MFA-enabled root accounts.

Monitoring and Alerts

- **Amazon GuardDuty:**
  - Enable GuardDuty to detect anomalous login attempts or unauthorised access.
- **CloudWatch Alarms:**
  - Monitor for:
    - Sign-ins from unusual geolocations.
    - Failed login attempts exceeding a predefined threshold.
    - Root account sign-ins.

Training and Awareness

- Train administrators to recognise and respond to alerts indicating suspicious login activity.
- Conduct regular incident response drills simulating unauthorised access scenarios.

2. DETECT

Indicators of Suspicious Login Activity

- **GuardDuty Findings:**
  - Alerts such as:
    - "UnauthorisedAccess: Root account login without MFA."
    - "Anomalous behavior: User login from a geolocation not previously seen."
- **CloudTrail Logs:**
  - Monitor ConsoleLogin events for:
    - Logins from unrecognised IPs or geolocations.
    - High frequency of failed login attempts.
    - Use of outdated or suspicious user agents.
- **AWS CloudWatch Metrics:**
  - Sudden spikes in login attempts or unauthorised actions.

Potential Triggers

- Sign-ins from unfamiliar IPs or locations.
- Multiple failed login attempts.
- Use of root account without MFA.

3. ANALYSE

Review Logs

- **CloudTrail Logs:**
  - Identify the source IP, geolocation and user agent.
  - Check associated actions following the login.
- **GuardDuty Findings:**
  - Analyse the severity and details of the findings to assess the threat level.
- **VPC Flow Logs:**
  - Investigate unusual outbound or inbound traffic patterns following suspicious logins.

Categorise the Incident

- Was the activity caused by:
  - Credential compromise?
  - Misconfiguration of IAM policies?
  - Malicious insider activity?

Assess Impact

- Determine if:
  - Unauthorised actions were performed post-login.
  - Sensitive data or configurations were accessed.

4. CONTAIN / ERADICATE

Containment Steps

- **Disable the User Account:**
  - Immediately suspend the compromised IAM user or role.
- **Revoke Active Sessions:**
  - Use aws iam delete-login-profile or similar commands to terminate active sessions.
- **Block Suspicious IPs:**
  - Update security group rules or WAF configurations to block malicious IPs.

Eradication Steps

- **Rotate Credentials:**
  - Reset the compromised user's password or access keys.
  - Rotate all credentials if root account access is suspected.
- **Audit IAM Roles and Policies:**
  - Remove overly permissive roles or policies.
- **Enable Geo-Restrictions:**
  - Restrict logins to trusted locations using SCPs or IAM policies.

5. RECOVER

Validate Recovery

- Verify that no unauthorised actions or access points remain.
- Ensure all credentials have been updated and MFA is enforced.

Restore Configurations

- Review and reinforce security configurations for IAM users and roles.
- Update monitoring and alerting rules to detect similar patterns earlier.

6. LESSONS LEARNT

Post-Incident Review

- Analyse the root cause of the incident:
  - Was it due to weak credentials, phishing or an unmonitored user?
- Evaluate the response timeline and identify bottlenecks.

Enhanced Security Practices

- Increase IAM audit frequency.
- Automate detection and response using AWS Lambda or third-party tools.
- Educate users on best practices for secure authentication.

AWS PLAYBOOK: RESPONDING TO EXCESSIVE FAILED LOGIN ATTEMPTS

1. PREPARATION

Baseline Security Configurations

- **Password Policy:**
  - Enforce strong password requirements (length, complexity and rotation).
  - Limit the number of password retries before locking accounts.
- **Multi-Factor Authentication (MFA):**
  - Mandate MFA for all users, especially for privileged accounts.
- **IAM Policies:**
  - Apply the principle of least privilege to minimise account misuse.
  - Restrict access to sensitive regions or actions using Service Control Policies (SCPs).
- **CloudTrail Logging:**
  - Enable AWS CloudTrail to log ConsoleLogin events across all regions.
  - Secure logs in an encrypted, private S3 bucket.
- **AWS Config Rules:**
  - Use rules to ensure MFA is enabled and strong password policies are enforced.

Monitoring and Alerts

- **Amazon GuardDuty:**
  - Enable GuardDuty to detect anomalous login behavior, such as brute force attempts.
- **CloudWatch Alarms:**
  - Monitor and set alarms for:
    - Excessive failed login attempts.
    - Repeated login attempts from the same IP or geolocation.

Training and Awareness

- Train users on password security and phishing awareness.
- Conduct simulated brute force attempts to test detection and response readiness.

2. DETECT

Indicators of Excessive Failed Logins

- **GuardDuty Findings:**
  - Alerts for brute force or unauthorised login attempts.
- **CloudTrail Logs:**

    - Analyse ConsoleLogin events for:
        - errorMessage: "Failed authentication"
        - Repeated login attempts from a single IP address.
- **CloudWatch Metrics:**
    - Sudden spikes in failed login attempts.

Potential Triggers

- Login attempts from unfamiliar geolocations or IPs.
- Login attempts outside of usual working hours.
- Failed login attempts targeting privileged IAM roles or users.

3. ANALYSE

Review Logs

- **CloudTrail Logs:**
    - Identify the source IPs and geolocations of the login attempts.
    - Determine which IAM user or role was targeted.
- **GuardDuty Findings:**
    - Assess the severity of brute force attempts.
- **VPC Flow Logs:**
    - Look for patterns of unusual traffic originating from suspicious IPs.

Categorise the Incident

- Was the activity caused by:
    - Credential stuffing or brute force attempts?
    - Exploitation of weak IAM policies?
    - Testing by a legitimate user or application?

Assess Impact

- Evaluate if the login attempts:
    - Resulted in successful unauthorised access.
    - Indicate a potential account compromise.

4. CONTAIN / ERADICATE

Containment Steps

- **Block Malicious IPs:**
    - Add suspicious IPs to WAF or security group rules for blocking.
- **Suspend Compromised Accounts:**

    - Temporarily disable the targeted IAM user or role.
- **Limit Login Attempts:**
    - Enforce account lockout after a specific number of failed attempts.

Eradication Steps

- **Credential Rotation:**
    - Reset passwords and access keys for compromised accounts.
- **Audit IAM Policies:**
    - Remove or restrict overly permissive roles or policies.
- **Enable Geo-Restrictions:**
    - Restrict logins to trusted geolocations.

5. RECOVER

Validate Recovery

- Ensure no unauthorised access occurred during the incident.
- Verify that malicious IPs are blocked and targeted accounts are secure.

Restore Configurations

- Update IAM policies to further restrict access.
- Enhance monitoring and alert thresholds to detect similar behavior earlier.

6. LESSONS LEARNT

Post-Incident Review

- Analyse the root cause:
    - Was it weak credentials, lack of MFA or unmonitored accounts?
- Assess response times and identify areas for improvement.

Enhanced Security Practices

- Regularly audit IAM policies and access controls.
- Increase the frequency of password and key rotation.
- Automate detection using AWS Lambda or third-party solutions.

AWS PLAYBOOK: RESPONDING TO MFA DISABLED FOR CRITICAL ACCOUNTS

1. PREPARATION

Baseline Security Configurations

- **Mandatory MFA for Critical Accounts:**
  - Enforce Multi-Factor Authentication (MFA) for all IAM users, especially for privileged accounts, using AWS IAM policies.
  - Ensure MFA is required for root user access to the AWS Management Console and API access.
- **IAM Policies:**
  - Create policies that restrict access for users without MFA enabled, specifically for actions related to critical resources (e.g., EC2, RDS, S3).
  - Implement AWS Organisations' Service Control Policies (SCPs) to ensure MFA is enforced at the organisational level.
- **CloudTrail Logging:**
  - Enable AWS CloudTrail for logging changes to MFA settings across all regions.
  - Monitor for any changes that disable or modify MFA settings for critical accounts.
  - Store CloudTrail logs securely in an encrypted, private S3 bucket.
- **AWS Config Rules:**
  - Use AWS Config to check if MFA is enabled for critical IAM users and enforce it.

Monitoring and Alerts

- **Amazon GuardDuty:**
  - Set up GuardDuty to detect suspicious activities such as unauthorised attempts to disable MFA on critical accounts.
- **CloudWatch Alarms:**
  - Set up alarms to detect when MFA is disabled or modified for critical accounts.
- **Custom IAM Policies Monitoring:**
  - Regularly audit IAM users with elevated privileges and ensure MFA is active.
  - Monitor changes to IAM roles or policies that could disable MFA.

Training and Awareness

- Regularly train administrators and security personnel on the importance of MFA for securing critical accounts.
- Run simulated phishing or social engineering exercises that attempt to disable MFA to test response procedures.

2. DETECT

Indicators of MFA Disabled for Critical Accounts

- **CloudTrail Logs:**
  - Check for UpdateMFADevice or DeleteMFADevice events in CloudTrail logs, which indicate changes to the MFA configuration.
  - Review any changes to IAM user or root user MFA configurations.
- **CloudWatch Metrics:**
  - Monitor for significant actions performed by accounts without MFA, such as changes to IAM policies or critical resource configurations.
- **GuardDuty Findings:**
  - Look for alerts on unusual behavior following the disabling of MFA, such as unauthorised access attempts or changes to sensitive AWS resources.
- **IAM Policies:**
  - Ensure policies reflect that critical accounts require MFA. If MFA is not enabled, it should trigger an alert.

3. ANALYSE

Review Logs

- **CloudTrail:**
  - Identify the IAM user who disabled MFA and the source IP address.
  - Check for any related suspicious activity or abnormal login patterns after MFA was disabled.
- **IAM Roles and Permissions:**
  - Review IAM roles and permissions to ensure they are restricted to users who need access.
  - Check for any roles that allow disabling or bypassing MFA.

Categorise the Incident

- Was the change due to:
  - Accidental misconfiguration?
  - Credential compromise?
  - Malicious insiders or external attackers?

Assess Impact

- Evaluate if disabling MFA allowed unauthorised access to critical accounts.
- Determine whether any sensitive or high-value resources were accessed without MFA.

4. CONTAIN / ERADICATE

Containment Steps

- **Re-enable MFA:**
  - Immediately re-enable MFA for the affected IAM users, especially for critical accounts.
  - Require the user to set up a new MFA device if the previous one was compromised.
- **Block Malicious Activity:**
  - Suspend or temporarily revoke the credentials of any compromised IAM users.
  - Block any IP addresses involved in disabling MFA or performing unauthorised activities.
- **Enforce MFA at the IAM Level:**
  - Implement strict IAM policies that prevent any user from disabling MFA without administrative approval.

Eradication Steps

- **Audit and Rotate Credentials:**
  - Rotate access keys and passwords for users whose MFA was disabled, as these could have been exposed.
- **Review and Strengthen IAM Policies:**
  - Review and update IAM roles and permissions to ensure no user can disable MFA on critical accounts.
  - Implement least-privilege access policies and restrict sensitive actions such as disabling MFA.
- **Account Compromise Investigation:**
  - Investigate if any account compromises occurred as a result of MFA being disabled.
  - Look for suspicious API calls or resource modifications linked to the user after MFA was disabled.

5. RECOVER

Validate Recovery

- Confirm that MFA is re-enabled for all critical accounts.
- Ensure that there is no residual malicious access and that all compromised credentials have been rotated.

Restore Configurations

- Reinforce IAM policies to prevent unauthorised changes to MFA settings.
- Audit all IAM users with elevated privileges to ensure that MFA is enforced.
- Ensure that CloudTrail logging and monitoring are set up to detect future changes to MFA configurations.

6. LESSONS LEARNT

Post-Incident Review

- Analyse the root cause of the incident:
    - Was MFA disabled due to a policy gap or lack of enforcement?
    - Were there any warning signs missed in monitoring?

Enhanced Security Practices

- **Automate MFA Enforcement:**
    - Use AWS Lambda to automate MFA enforcement and alerting for any critical account configuration changes.
- **Frequent Security Audits:**
    - Increase the frequency of security audits to ensure that MFA policies are always enforced across critical accounts.
- **Implement Backup MFA Devices:**
    - Ensure backup MFA devices are available for administrators in case of hardware failures or issues with primary MFA devices.

AWS PLAYBOOK: RESPONDING TO PUBLIC ACCESS TO S3 BUCKETS

1. PREPARATION

Baseline Security Configurations

- **Block Public Access Settings:**
  - Enable S3 Block Public Access at the bucket and account levels to prevent accidental exposure.
  - Ensure that the "Block all public access" setting is enabled for all S3 buckets in your environment.
- **IAM Policies:**
  - Implement IAM policies that restrict the ability to modify S3 bucket permissions to a limited set of users.
  - Use least-privilege access policies for users interacting with S3 buckets to minimise exposure risk.
- **CloudTrail Logging:**
  - Enable CloudTrail logging for S3 bucket access events.
  - Store CloudTrail logs in a secure, encrypted S3 bucket for analysis and auditing.
- **S3 Bucket Logging:**
  - Enable access logging on all S3 buckets to track who accesses your data and from where.
- **AWS Config:**
  - Use AWS Config to ensure that S3 Block Public Access settings are applied to all S3 buckets.
  - Set up AWS Config rules to trigger alerts if public access permissions are granted to buckets.

Monitoring and Alerts

- **CloudWatch Alarms:**
  - Set up alarms to notify administrators of any changes to public access settings on S3 buckets.
  - Monitor for unusual access patterns such as traffic from untrusted IP addresses or spikes in access frequency.
- **GuardDuty:**
  - Enable GuardDuty to detect suspicious activities such as data exfiltration or unauthorised access to publicly accessible S3 buckets.

Training and Awareness

- Train administrators and developers on proper S3 bucket security practices, emphasising the risks of public access.

- Conduct regular security audits to ensure compliance with S3 bucket access control policies.

2. DETECT

Indicators of Public Access to S3 Buckets

- **CloudTrail Logs:**
  - Check CloudTrail logs for any changes to the bucket's ACLs (Access Control Lists) or bucket policy that could expose the bucket publicly.
  - Review any PutBucketPolicy or PutBucketAcl events for public access configuration changes.
- **S3 Access Logs:**
  - Review S3 access logs for signs of unauthorised or unexpected public access to sensitive data.
- **AWS Config:**
  - Use AWS Config to detect any non-compliance with public access settings for S3 buckets.
- **GuardDuty Findings:**
  - Look for GuardDuty findings indicating suspicious access to S3 buckets or anomalous traffic patterns.
- **CloudWatch Metrics:**
  - Look for unexpected spikes in traffic to your S3 buckets, especially from unfamiliar geolocations or IP addresses.

3. ANALYSE

Review Logs

- **CloudTrail Logs:**
  - Identify which user or service made the change to the bucket policy or ACL that allowed public access.
  - Check for any unauthorised access to data following the exposure of the bucket.
- **S3 Bucket Configuration:**
  - Review the bucket's permissions (e.g., ACLs, bucket policies and IAM policies) to determine how public access was granted.
  - Check the S3 console for any publicly accessible objects or buckets.
- **VPC Flow Logs:**
  - Review VPC flow logs to detect any unexpected data transfers from the exposed bucket.

Categorise the Incident

- Was the public access to the S3 bucket caused by:
    - Misconfiguration of access control settings?
    - Unauthorised changes to bucket policy by a compromised account?
    - Misuse of IAM permissions allowing unauthorised users to modify the S3 bucket configuration?

Assess Impact

- Determine if any sensitive data was exposed publicly due to misconfigured permissions.
- Assess the scale of exposure by reviewing the number of files accessed and whether any confidential data was accessed or downloaded.

4. CONTAIN / ERADICATE

Containment Steps

- **Revert Access Permissions:**
    - Immediately disable public access by using the “Block all public access” setting in the S3 bucket configuration.
    - Review and update the bucket's ACLs, policies and IAM roles to prevent public access.
- **Isolate Compromised Accounts:**
    - Revoke any IAM user or role access that made unauthorised changes to S3 bucket permissions.
    - Suspend or terminate any compromised IAM credentials involved in the misconfiguration.
- **Monitor for Ongoing Exposure:**
    - Set up alerts to monitor for any changes to S3 bucket configurations or permissions to ensure that public access does not get re-enabled.

Eradication Steps

- **Audit and Rotate Credentials:**
    - Rotate IAM credentials for users who had the ability to modify S3 permissions.
- **Review and Enforce Access Control Policies:**
    - Enforce IAM policies that restrict the ability to modify S3 permissions to a limited set of trusted administrators.
- **Incident Response:**
    - Investigate whether any sensitive data was accessed, exfiltrated or leaked during the time the bucket was publicly accessible.

5. RECOVER

Validate Recovery

- Ensure all S3 buckets are securely configured, with public access completely blocked.
- Confirm that no unauthorised access occurred to any sensitive data stored in the exposed buckets.

Restore Configurations

- Reinforce the use of “Block all public access” settings for all S3 buckets.
- Update IAM policies to prevent unauthorised users from changing S3 access controls.
- Set up continuous monitoring to detect any future changes to bucket permissions.

6. LESSONS LEARNT

Post-Incident Review

- Analyse the root cause of the incident:
    - Was the exposure due to a lack of awareness or training on S3 security settings?
    - Was it caused by a misconfiguration in IAM policies or automated workflows?

Enhanced Security Practices

- **Automation for Access Control:**
    - Use AWS Lambda to automate monitoring and enforcement of public access settings on S3 buckets.
- **Strengthen Monitoring and Alerts:**
    - Implement more frequent checks on S3 bucket permissions and integrate them with automated response mechanisms.
- **Review Access Control Policies:**
    - Regularly audit and update IAM policies to ensure that only authorised users can modify critical access settings on S3 buckets.

AWS PLAYBOOK: RESPONDING TO DATA EXFILTRATION

1. PREPARATION

Baseline Security Configurations

- **Enable GuardDuty:**
  - Enable AWS GuardDuty to detect suspicious data exfiltration activities, such as unusual network traffic or communications with known malicious IPs.
- **IAM Policies:**
  - Restrict permissions for accessing sensitive data to only those users who absolutely need it (Principle of Least Privilege).
  - Use IAM roles and groups to manage permissions for different services and data access.
  - Enforce multi-factor authentication (MFA) for sensitive accounts or those with access to critical resources.
- **CloudTrail Logging:**
  - Ensure AWS CloudTrail is enabled for all regions to capture API calls made to AWS resources.
  - Store CloudTrail logs in a private, encrypted S3 bucket for forensic analysis.
- **VPC Flow Logs:**
  - Enable VPC Flow Logs to track network traffic and detect any unusual data transfer patterns.
- **AWS Config:**
  - Use AWS Config to ensure security configurations, such as encryption and logging, are maintained and compliant with internal policies.

Monitoring and Alerts

- **CloudWatch Alarms:**
  - Set up CloudWatch alarms to monitor for unusual data transfer volumes or sudden spikes in outbound network traffic that could indicate data exfiltration.
  - Monitor for large-scale data movement, especially to external locations or untrusted regions.
- **GuardDuty:**
  - Configure GuardDuty to send alerts for signs of data exfiltration, such as communication with known bad IP addresses, unusual access patterns or anomalies in network activity.
- **Data Loss Prevention (DLP) Tools:**
  - Use AWS Macie to monitor and protect sensitive data, including personal identifiable information (PII), credit card details or other critical data from exfiltration attempts.

Training and Awareness

- Provide security awareness training to all employees to recognise signs of data exfiltration, such as unauthorised data downloads, suspicious email attachments or unusual network activity.
- Regularly simulate data exfiltration scenarios as part of tabletop exercises or red team assessments.

2. DETECT

Indicators of Data Exfiltration

- **GuardDuty Findings:**
  - Alerts for data moving to known malicious IP addresses.
  - Unusual API calls or network traffic patterns suggesting data movement to external destinations.
  - Findings that indicate suspicious or unauthorised access to sensitive data (e.g., database dumps, large file downloads).
- **CloudTrail Logs:**
  - Review CloudTrail logs for suspicious GetObject, PutObject or other S3 data access API calls, especially from unexpected locations or accounts.
  - Monitor AssumeRole or Sts:AssumeRole events that could indicate an attacker escalating privileges or accessing critical resources.
- **VPC Flow Logs:**
  - Look for unusual outbound traffic, especially to untrusted IP addresses, regions or suspicious ports.
  - Identify large volumes of traffic leaving the VPC.
- **S3 Access Logs:**
  - Review S3 bucket logs to detect unauthorised downloads or uploads, especially from unapproved IP addresses.
- **Macie Findings:**
  - AWS Macie may detect sensitive data being transferred outside of your environment. Monitor Macie findings for data exfiltration attempts.

Potential Triggers

- **Unusual Access Behavior:**
  - Anomalous access to critical data or services by unauthorised or unfamiliar users.
  - Large numbers of GetObject or PutObject API calls within a short period, especially for large files.
- **Unexpected Traffic Patterns:**
  - Unusual volumes of data being transferred out of AWS to external locations or unfamiliar IP addresses.

- Large-scale database or S3 bucket downloads.

3. ANALYSE

Review Logs

- **CloudTrail Logs:**
  - Identify the origin of data access requests, including the IAM user, the source IP and the service or resource being accessed.
  - Look for unusual patterns, such as large volumes of data being moved or access from unauthorised IPs or geolocations.
- **VPC Flow Logs:**
  - Review network traffic logs for abnormal traffic leaving your network, especially to unapproved or external IP addresses.
  - Look for high traffic volumes over unusual ports or protocols.
- **Macie Findings:**
  - Review Macie logs for signs of PII or sensitive data being accessed or transferred to an external location.
- **S3 and S3 Access Logs:**
  - Review logs for large-scale downloads from critical S3 buckets, particularly for sensitive files like databases or large amounts of personal data.

Categorise the Incident

- **Was the exfiltration caused by:**
  - A compromised IAM user or role?
  - Misconfiguration of access permissions?
  - Malicious insider activity?

Assess Impact

- Assess the volume of data that has been exfiltrated and the types of data (e.g., PII, intellectual property, financial data).
- Investigate the destination of the exfiltrated data and whether it was accessed by an external, unauthorised party.
- Determine if any data was leaked or accessed by competitors or other malicious entities.

4. CONTAIN / ERADICATE

Containment Steps

- **Revoke Access:**

    - Immediately revoke the credentials of any users or IAM roles involved in the exfiltration, including rotating keys, passwords and terminating sessions.
- **Isolate Resources:**
    - If the exfiltration involves specific services or instances, isolate those from the network by removing their internet access or shutting them down.
- **Block External Communication:**
    - Use VPC security groups and NACLs (Network ACLs) to block outgoing traffic to suspicious IP addresses or external destinations.

Eradication Steps

- **Terminate or Suspend Affected Resources:**
    - Disable or terminate instances that were involved in the exfiltration (e.g., EC2 instances).
- **Patch Vulnerabilities:**
    - Ensure that all vulnerabilities or misconfigurations (e.g., overly permissive IAM policies, open ports) are corrected to prevent further exfiltration attempts.
- **Review and Update Security Controls:**
    - Review IAM roles and permissions to ensure they follow the principle of least privilege.
    - Audit and tighten CloudTrail, S3 and VPC settings to prevent future exfiltration.

5. RECOVER

Validate Recovery

- Confirm that no further exfiltration activity is occurring by reviewing logs and monitoring network traffic.
- Ensure that IAM credentials are rotated and access controls are tightened to prevent unauthorised data access.

Restore Configurations

- Reinforce IAM policies and roles to enforce least-privilege access.
- Implement stricter VPC traffic monitoring and controls.
- Update CloudTrail, S3 and GuardDuty configurations to enhance detection and response to similar incidents.

6. LESSONS LEARNT

Post-Incident Review

- **Root Cause Analysis:**
    - Determine how the exfiltration was able to occur—was it a misconfiguration, a compromised account or a vulnerability that was exploited?
- **Identify Gaps in Detection:**
    - Were the existing monitoring systems (CloudWatch, GuardDuty, etc.) sufficient to detect and alert on the data exfiltration? What can be improved?

Enhanced Security Practices

- **Automation:**
    - Implement automated alerts or Lambda functions to prevent or respond to data exfiltration attempts in real-time.
- **Tighten Access Controls:**
    - Implement stricter IAM policies and use multi-factor authentication (MFA) on sensitive accounts.
- **Regular Audits and Monitoring:**
    - Conduct regular audits of IAM permissions, network configurations and data access controls.
    - Continuously monitor VPC Flow Logs and CloudTrail logs for unusual outbound traffic.

AWS PLAYBOOK: RESPONDING TO UNUSUAL API CALLS

1. PREPARATION

Baseline Security Configurations

- **Enable CloudTrail:**
  - Ensure AWS CloudTrail is enabled across all regions to log API activity for all services in your environment.
  - Store CloudTrail logs in an encrypted, private S3 bucket for security and auditing purposes.
  - Set up CloudTrail event filtering to focus on critical APIs, such as IAM, EC2 and S3.
- **IAM Policies:**
  - Follow the principle of least privilege to limit access to API actions to only those who require it.
  - Regularly review and update IAM roles and policies to ensure only the necessary permissions are granted.
- **Service Control Policies (SCPs):**
  - Use SCPs in AWS Organisations to limit the actions that can be taken across accounts and regions, minimising exposure to unusual API calls.
- **VPC Flow Logs:**
  - Enable VPC Flow Logs to capture network traffic and help detect unusual API calls originating from or affecting network resources.

Monitoring and Alerts

- **CloudWatch Alarms:**
  - Set CloudWatch alarms for unexpected API calls, such as those from unusual IAM users or roles.
  - Configure CloudWatch to monitor high-risk actions such as iam:CreateUser, ec2:TerminateInstances or s3:DeleteBucket.
- **GuardDuty:**
  - Enable GuardDuty to detect suspicious API activity and issues such as privilege escalation, API abuse or usage of compromised credentials.
- **AWS Config:**
  - Use AWS Config to ensure configurations are compliant with security policies and detect changes in resource configurations that may be triggered by unusual API calls.

Training and Awareness

- Train administrators and security teams to recognise the signs of suspicious or unusual API calls.

- Conduct regular exercises on responding to different API-based incidents, including changes in critical resources or unauthorised access to services.

2. DETECT

Indicators of Unusual API Calls

- **CloudTrail Logs:**
  - Look for API calls made by unauthorised or unusual IAM users, roles or AWS accounts.
  - Review CloudTrail logs for API actions that are outside of normal usage patterns (e.g., a sudden spike in CreateBucket or DeleteBucket API calls).
  - Look for API calls originating from unexpected geolocations or IP addresses.
- **GuardDuty Findings:**
  - GuardDuty can detect unusual API activity, such as:
    - **Credential stuffing attacks**: Multiple failed authentication attempts followed by a successful API call.
    - **Privilege escalation**: API calls that modify user permissions or roles.
    - **Anomalous login activity**: Access to the AWS environment from unusual regions or unknown IP addresses.
- **CloudWatch Metrics:**
  - Spikes in API calls, especially high-risk operations such as TerminateInstances or DeleteBucket.
  - Patterns indicating an automated or scripted attack, like repetitive or high-frequency API requests.
- **IAM Role Anomalies:**
  - Unauthorised changes in IAM policies, roles or permissions.
  - API calls made by a role or user with elevated privileges that shouldn't have access.

Potential Triggers

- **Unusual Locations or Times:**
  - API calls from unfamiliar IP addresses or regions, particularly if they coincide with periods of inactivity.
- **API Rate Limits:**
  - Unusual API call rates (e.g., bursts of requests) that may indicate an attempt to probe or exploit AWS services.
- **Unauthorised Resource Modifications:**
  - API calls affecting critical resources, like modifying or deleting EC2 instances, changing security groups or modifying S3 bucket policies.

3. ANALYSE

Review Logs

- **CloudTrail Logs:**
  - Review the event details to identify the IAM user or role that initiated the suspicious API calls.
  - Examine the API actions performed, their parameters and the resources affected.
  - Check for unusual patterns, such as repeated API calls from a single user or a change in a pattern of activity.
- **VPC Flow Logs:**
  - Cross-reference VPC flow logs to identify any network traffic that correlates with the suspicious API activity.
  - Look for outbound traffic to unknown or suspicious external IP addresses.
- **IAM Logs and Policies:**
  - Review changes to IAM roles or policies and verify whether the API call was made by a compromised or elevated privilege role.

Categorise the Incident

- **Was the unusual API call caused by:**
  - A misconfigured IAM policy or role escalation?
  - Compromised credentials?
  - Malicious insiders?
- **Determine the Extent of Impact:**
  - Evaluate whether the unusual API calls resulted in unauthorised access to resources or changes to the environment (e.g., new EC2 instances launched, data moved to S3).

4. CONTAIN / ERADICATE

Containment Steps

- **Revoke Access:**
  - Immediately revoke the credentials of any compromised IAM user or role, rotating passwords and keys if necessary.
  - Block any malicious IP addresses or geolocations from which the suspicious API calls originated.
- **Limit Permissions:**
  - Reduce the permissions of users or roles associated with unusual API calls and apply tighter service control policies (SCPs) to limit access to critical services.
- **Isolate Affected Resources:**
  - If resources were affected by unusual API calls (e.g., EC2 instances or S3 buckets), isolate or shut them down to prevent further damage.

Eradication Steps

- **Audit and Remediate:**
  - Audit IAM roles, policies and access keys for any suspicious configurations.
  - Terminate any compromised or rogue instances, revoke access tokens and delete any unauthorised resources.
- **Review and Harden Configurations:**
  - Reinforce security configurations by reviewing IAM roles, setting up stricter policies and improving logging and monitoring.
- **Update Security Policies:**
  - Review and update security policies to restrict access to sensitive resources.
  - Consider implementing more granular permissions and using least-privilege principles across all IAM roles.

5. RECOVER

Validate Recovery

- **Ensure no further unusual API calls:**
  - Use CloudWatch to ensure that the unusual API call activity has ceased and that the environment is stable.
- **Check for Residual Compromise:**
  - Conduct a full security review to ensure that no further issues remain. Rotate any compromised access credentials, including IAM keys, passwords and secrets.

Restore Configurations

- **Reinforce Access Controls:**
  - Ensure that IAM permissions, security groups and SCPs are updated to restrict access to sensitive resources.
- **Update CloudTrail and Monitoring:**
  - Update CloudTrail filters to ensure that all critical API actions are captured.
  - Set up additional monitoring for high-risk API calls to catch potential issues earlier in the future.

6. LESSONS LEARNT

Post-Incident Review

- **Root Cause Analysis:**
  - Identify how the unusual API calls were able to be initiated—was it a result of misconfigured IAM policies, lack of monitoring or a compromised account?

- **Improve Detection and Response:**
    - Ensure that automated detection systems like GuardDuty and CloudWatch are tuned to detect such activities sooner.
    - Consider adding more granular API call logging for high-risk services.

Enhanced Security Practices

- **Automation:**
    - Use AWS Lambda to automatically respond to suspicious API calls, such as terminating instances or blocking IP addresses.
- **Implement Rate-Limiting:**
    - Implement rate-limiting or throttling for sensitive API calls to reduce the potential for exploitation.
- **Frequent Audits:**
    - Conduct regular audits of IAM roles and permissions to ensure compliance with security best practices.
- **Better Alerting and Monitoring:**
    - Increase the frequency and granularity of CloudWatch alarms and GuardDuty alerts for critical API calls.

AWS PLAYBOOK: RESPONDING TO ROOT ACCOUNT ACTIVITY

1. PREPARATION

Baseline Security Configurations

- **Enable Multi-Factor Authentication (MFA) for the Root Account:**
  - Ensure that MFA is enabled on the root account to add an additional layer of security against unauthorised access.
- **Enable CloudTrail:**
  - Ensure AWS CloudTrail is enabled across all regions to capture all activity performed by the root account.
  - Store CloudTrail logs in a secure, encrypted S3 bucket to track and audit any root account activity.
- **IAM Policies:**
  - Limit the use of the root account and ensure it is only used for essential administrative tasks. Create IAM users with specific permissions for daily operational tasks.
  - Review and implement Service Control Policies (SCPs) to restrict the use of the root account in AWS Organisations.
- **GuardDuty:**
  - Enable Amazon GuardDuty to monitor for suspicious root account activity, such as login attempts from unfamiliar IP addresses or geolocations.
- **AWS Config:**
  - Set up AWS Config to monitor and record changes to your AWS environment, paying particular attention to changes made by the root account.

Monitoring and Alerts

- **CloudWatch Alarms:**
  - Set up CloudWatch alarms to monitor for root account login events or critical actions taken by the root user, such as changes to IAM roles, security groups or billing settings.
- **GuardDuty Findings:**
  - Configure GuardDuty to alert on root account usage, especially for high-risk activities like modifying IAM policies or terminating instances.
- **CloudTrail Alerts:**
  - Set up real-time alerts for sensitive API calls made by the root account, such as iam:CreateUser, iam:DeleteUser or ec2:TerminateInstances.

Training and Awareness

- **Administrator Training:**

    - Educate AWS administrators on the importance of limiting root account usage and securing it with MFA.
    - Train administrators on how to respond to suspicious root account activity, including how to lock or revoke access when necessary.

2. DETECT

Indicators of Root Account Activity

- **CloudTrail Logs:**
    - Monitor CloudTrail logs for any ConsoleLogin events where the root account is used. Check for login attempts from unfamiliar IP addresses or geographic locations.
    - Review logs for sensitive API calls made by the root account, such as iam:CreateUser, iam:DeleteUser or changes to billing information.
- **GuardDuty Findings:**
    - GuardDuty can detect:
        - **Suspicious login attempts** using the root account from unfamiliar IPs.
        - **Unusual actions**, such as modifying IAM roles or security settings.
        - **Potential privilege escalation**, where the root account's credentials are used to grant elevated privileges to other users.
- **CloudWatch Metrics:**
    - Detect sudden spikes in API calls made by the root account, especially during non-peak hours or from unapproved regions.
- **IAM Role Anomalies:**
    - Unexpected changes to IAM policies or roles, which may suggest the root account was used to escalate privileges.

Potential Triggers

- **Root Account Login from Unusual Locations:**
    - Logins from unknown or unapproved IP addresses or geographic locations.
- **Sensitive Actions Taken by Root:**
    - Creation or deletion of IAM users or roles, modification of security groups or changes to billing settings.
- **Privileged Resource Modifications:**
    - Changes to sensitive resources such as VPC settings, security configurations or critical EC2 instances.

3. ANALYSE

Review Logs

- **CloudTrail Logs:**
    - Examine the CloudTrail logs to determine the origin of the root account activity. Look for:
        - The source IP address, geographic location and time of the login.
        - The sequence of actions performed by the root account.
        - If any IAM roles or user permissions were modified following root account actions.
- **VPC Flow Logs:**
    - Review any associated VPC flow logs to correlate network activity with the root account actions.
    - Look for any unusual outbound traffic from resources modified by the root account.
- **IAM Logs:**
    - Review IAM activity logs to determine if the root account modified IAM policies or created/deleted any users or roles.

Categorise the Incident

- **Determine the cause of the activity:**
    - Was the root account activity legitimate or a result of compromise?
    - Was the activity caused by internal or external threat actors?
- **Evaluate Potential Impact:**
    - Did the root account activity result in any significant changes to your AWS environment, such as privilege escalation, data breaches or resource modifications?

4. CONTAIN / ERADICATE

Containment Steps

- **Disable Root Account Access:**
    - Temporarily disable the root account by resetting its password and requiring MFA for any subsequent logins.
    - Revoke or rotate any compromised credentials if the root account credentials were exposed.
- **Revoke Access to Affected Resources:**
    - If the root account was used to modify or access sensitive resources, immediately revoke access to those resources (e.g., modify IAM policies, security groups or terminate instances).
- **Block Suspicious IPs:**
    - Block any IP addresses or geolocations associated with unauthorised root account access.

Eradication Steps

- **Terminate or Revert Unauthorised Changes:**
    - Revert any changes made by the root account that are deemed suspicious or unauthorised, such as unauthorised IAM user creation, security group modifications or billing changes.
    - Terminate any rogue instances or services launched by the root account.
- **Audit and Remediate IAM Policies:**
    - Review all IAM roles and permissions and ensure that they are configured according to best practices.
    - Implement stricter IAM policies and service control policies (SCPs) to prevent unauthorised access.

5. RECOVER

Validate Recovery

- **Confirm Root Account Activity Has Ceased:**
    - Ensure that all root account actions have been logged and contained. Re-enable the root account with MFA if necessary.
- **Check for Residual Compromise:**
    - Review all IAM users, roles and service accounts to ensure that no other accounts were compromised as a result of the root account activity.

Restore Configurations

- **Reinforce Access Control:**
    - Ensure that IAM roles and permissions are updated to reflect stricter access control and limit root account use.
- **Update CloudTrail and Monitoring:**
    - Review CloudTrail logs to ensure that root account activity is being captured and implement new alerts for root account usage.
    - Set up more granular monitoring for root account actions to detect any future suspicious activity.

6. LESSONS LEARNT

Post-Incident Review

- **Root Cause Analysis:**
    - Determine how the root account was accessed and what vulnerabilities allowed for the activity (e.g., lack of MFA, compromised credentials).
    - Identify if there were gaps in logging or monitoring that allowed the incident to go unnoticed.
- **Update Security Procedures:**

    - Update incident response procedures and training to cover root account activity.
    - Ensure that future incidents involving the root account can be detected and mitigated more quickly.

Enhanced Security Practices

- **Automate Detection:**
    - Use AWS Lambda or other automation tools to automatically respond to suspicious root account activity, such as disabling the account or blocking IP addresses.
- **Increase MFA Enforcement:**
    - Consider enforcing MFA for all AWS accounts, especially for users with access to critical resources.
- **Regular IAM Audits:**
    - Conduct regular IAM audits to ensure that no user or role has more permissions than necessary.
- **Implement Least-Privilege Access:**
    - Apply the principle of least privilege to all IAM roles, ensuring that the root account is only used when absolutely necessary.

AWS PLAYBOOK: RESPONDING TO AWS CONFIG CHANGES

1. PREPARATION

Baseline Security Configurations

- **Enable AWS Config:**
  - Enable AWS Config to monitor and record all configuration changes to AWS resources across all regions.
  - Ensure AWS Config is capturing changes to key resources, such as EC2 instances, security groups, IAM roles and VPC settings.
- **Set Up CloudTrail:**
  - Ensure AWS CloudTrail is enabled for all regions and integrated with AWS Config to capture detailed records of configuration changes and user activity.
- **Service Control Policies (SCPs):**
  - Apply SCPs in AWS Organisations to restrict certain AWS Config changes, such as modifying or disabling configuration recording.
- **CloudWatch Alarms:**
  - Set up CloudWatch alarms to detect any unauthorised or unexpected changes to critical AWS resources, such as IAM roles, security settings or VPC configurations.
- **IAM Policies:**
  - Implement least privilege principles by defining specific IAM roles and policies that restrict who can modify AWS Config settings and access AWS resources.
  - Restrict AWS Config modification permissions to administrators or designated security personnel.

Monitoring and Alerts

- **AWS Config Rules:**
  - Define and enforce AWS Config rules to ensure that configuration settings comply with security best practices.
  - Set up alerts for non-compliant resources or deviations from approved configurations.
- **CloudWatch Alarms:**
  - Set up CloudWatch alarms to detect and notify security teams of changes to configuration settings or resource configurations in real time.
- **GuardDuty:**
  - Enable GuardDuty to identify any malicious or suspicious activity that may trigger or coincide with unauthorised AWS Config changes.

Training and Awareness

- **Administrator Training:**
    - Train administrators on the importance of configuration management and the security implications of AWS Config changes.
    - Provide training on responding to unauthorised or suspicious configuration changes in AWS.

2. DETECT

Indicators of AWS Config Changes

- **AWS Config Logs:**
    - Monitor AWS Config logs for unauthorised changes to the configuration of critical resources, such as IAM policies, VPC settings, security groups or EC2 instances.
    - Look for configuration changes made by unusual or unexpected users, IP addresses or geolocations.
- **CloudTrail Logs:**
    - Review CloudTrail logs to detect who made the configuration changes and what actions were performed. Look for specific API calls such as PutConfigRule, DeleteConfigRule or StopConfigurationRecorder.
- **GuardDuty Findings:**
    - GuardDuty can alert on suspicious activity that may coincide with AWS Config changes, such as unexpected API calls or potential privilege escalation.
- **CloudWatch Metrics:**
    - Set CloudWatch metrics to identify unexpected spikes in configuration changes or unauthorised updates to resource configurations.

Potential Triggers

- **Unauthorised Configuration Changes:**
    - Changes to IAM roles, VPC settings, security groups, EC2 instances or S3 bucket configurations made by unauthorised users.
- **Configuration Drift:**
    - Sudden or unexpected deviations from approved configurations, such as changes to security group rules, network configurations or other critical settings.
- **Privilege Escalation:**
    - Changes made by users who are not authorised to modify certain configurations, especially IAM policies and roles.

3. ANALYSE

Review Logs

- **CloudTrail Logs:**
    - Investigate who initiated the AWS Config changes and their associated IP address.
    - Check for any other related changes to resources or permissions that may indicate a broader attack or unauthorised activity.
    - Look for patterns of changes, such as modifications to security groups, IAM policies or resource configurations.
- **AWS Config History:**
    - Review the history of AWS Config changes for any deviations from the baseline configurations, such as security settings, network configurations and IAM roles.
- **CloudWatch Logs:**
    - Analyse CloudWatch logs to see if the configuration changes were part of a larger set of unexpected activities, such as network traffic spikes or unauthorised API calls.

Categorise the Incident

- **Determine the cause of the changes:**
    - Was the change authorised and part of a legitimate configuration update or was it malicious or accidental?
    - Was there an escalation of privileges that allowed unauthorised users to make configuration changes?
- **Assess Impact:**
    - Check for any impact of the configuration changes on the security posture of your environment.
    - Did the changes allow for unauthorised access to sensitive resources, such as misconfigured security groups or IAM roles?

4. CONTAIN / ERADICATE

Containment Steps

- **Isolate the Affected Resources:**
    - If unauthorised changes have been made to critical resources, isolate the affected instances or services by changing their security group or network configurations.
- **Revoke Unauthorised Access:**
    - Revoke the IAM credentials used to make the unauthorised AWS Config changes.
    - Modify IAM roles and policies to prevent further unauthorised configuration changes.
- **Review and Block Suspicious IPs:**

    - If the changes originated from suspicious or unknown IP addresses, block them at the VPC or security group level.

Eradication Steps

- **Revert Unauthorised Changes:**
    - Revert all unauthorised configuration changes made to critical AWS resources.
    - Restore configurations to their secure, compliant states, especially for IAM policies, VPC settings and security groups.
- **Audit IAM and Access Policies:**
    - Review IAM roles, policies and permissions to ensure they are set according to least privilege principles. Address any misconfigurations that allowed unauthorised changes.
- **Review AWS Config Rules:**
    - Ensure that AWS Config rules are updated to detect similar unauthorised changes in the future. Tighten the rules to prevent future drift from approved configurations.

5. RECOVER

Validate Recovery

- **Confirm Changes Have Been Reversed:**
    - Ensure all unauthorised configuration changes have been reversed and that the AWS environment is back to a secure, compliant state.
- **Check for Residual Impact:**
    - Review logs and monitoring alerts to confirm there are no remaining impacts from the unauthorised changes, such as compromised resources or services.

Restore Configurations

- **Reinforce Configuration Management:**
    - Revisit AWS Config settings to ensure that configuration rules are working as expected and capturing changes accurately.
    - Implement stricter access controls around configuration changes and who can modify AWS Config settings.
- **Strengthen Monitoring and Alerts:**
    - Ensure that CloudTrail, CloudWatch and GuardDuty are fully integrated and can provide real-time alerts for unauthorised AWS Config changes.
    - Update alerting thresholds and response workflows to handle similar incidents more quickly in the future.

6. LESSONS LEARNT

Post-Incident Review

- **Root Cause Analysis:**
  - Conduct a thorough analysis of how the unauthorised AWS Config changes were made and identify any gaps in security controls or monitoring that allowed the incident to occur.
  - Evaluate if there were any weaknesses in IAM roles, security groups or AWS Config rules that were exploited.
- **Update Security Practices:**
  - Update incident response playbooks to incorporate detection and response to unauthorised AWS Config changes.
  - Strengthen procedures for detecting configuration drift or other signs of unauthorised changes.

Enhanced Security Practices

- **Automate Detection and Response:**
  - Leverage AWS Lambda to automatically remediate unauthorised changes, such as reverting configurations or isolating compromised resources.
- **Increase Frequency of IAM Audits:**
  - Perform regular audits of IAM roles and permissions to ensure least privilege and reduce the risk of privilege escalation.
- **Continuous Monitoring:**
  - Implement continuous monitoring for AWS Config changes, ensuring that any deviation from secure configurations is flagged immediately.

AWS PLAYBOOK: RESPONDING TO CLOUDTRAIL LOGS DISABLED

1. PREPARATION

Baseline Security Configurations

- **Ensure CloudTrail is Always Enabled:**
  - CloudTrail should be enabled in all AWS accounts and regions to monitor and log all API activity. Ensure that multi-region logging is turned on to capture events across all AWS services in your organisation.
- **CloudTrail Integrity Monitoring:**
  - Enable CloudTrail integrity monitoring to detect and alert if CloudTrail logs are tampered with or if logging is disabled.
- **Log Storage:**
  - Store CloudTrail logs in an encrypted S3 bucket, with proper access controls, to prevent unauthorised access or tampering.
  - Use lifecycle policies to retain logs for an appropriate duration, ensuring compliance with internal security policies.
- **IAM Policies:**
  - Implement strict IAM policies for controlling access to CloudTrail configuration and logs. Restrict permissions to CloudTrail management only to a small group of trusted administrators.
  - Use Service Control Policies (SCPs) to prevent disabling CloudTrail across all AWS accounts in AWS Organisations.

Monitoring and Alerts

- **CloudWatch Alarms:**
  - Set up CloudWatch Alarms to alert when CloudTrail logging is turned off or if CloudTrail configuration changes occur.
  - Create alarms for suspicious activity, such as deletion or modification of CloudTrail logs.
- **GuardDuty:**
  - Enable GuardDuty to monitor for unusual or unauthorised activity that may coincide with CloudTrail being disabled, such as API calls to disable logging or changes in IAM policies that could affect CloudTrail configuration.
- **CloudTrail Configuration Compliance:**
  - Use AWS Config to monitor CloudTrail settings and ensure compliance with configuration rules that mandate CloudTrail logging is always enabled.

Training and Awareness

- **Administrator Training:**

    - Train administrators on the importance of CloudTrail for auditing and security monitoring and the procedures to follow if CloudTrail logs are disabled.
- **Incident Response Drills:**
    - Conduct regular incident response drills, including scenarios where CloudTrail is disabled, to ensure that the team can react quickly and appropriately.

2. DETECT

Indicators of CloudTrail Logs Disabled

- **CloudTrail Logs:**
    - Investigate if CloudTrail logs stop appearing in the S3 bucket or if there is an unexpected gap in log data, indicating that logging was disabled.
- **CloudTrail Event History:**
    - Look for StopLogging or DeleteTrail API calls in CloudTrail's event history, which could indicate that logging has been disabled.
- **GuardDuty Findings:**
    - GuardDuty may detect suspicious activities or abnormal API calls, such as disabling CloudTrail or changes to IAM roles that can impact logging functionality.
- **CloudWatch Alarms:**
    - CloudWatch alarms configured to monitor CloudTrail logging status may trigger if logging is disabled or if there are unexpected gaps in logs.

Potential Triggers

- **Intentional Disabling:**
    - If CloudTrail logging is disabled intentionally, it may be for troubleshooting purposes or a configuration mistake.
- **Malicious Activity:**
    - An attacker may disable CloudTrail logs to hide malicious activities or avoid detection.
- **Accidental Misconfigurations:**
    - A misconfiguration or accidental change in IAM policies or CloudTrail settings could result in logging being disabled.

3. ANALYSE

Review Logs

- **CloudTrail:**

    - Investigate if any user or service has executed the StopLogging or DeleteTrail API calls. Identify who performed these actions and from which IP address.
- **IAM Logs:**
    - Review CloudTrail logs for any changes to IAM policies or roles that might have granted permissions to disable CloudTrail. Check for unauthorised changes to IAM policies, roles or permissions that could have allowed the disabling of CloudTrail logs.
- **CloudWatch Logs:**
    - Review CloudWatch logs to identify patterns or suspicious activity leading up to the disabling of CloudTrail logs.
- **S3 Bucket Logs:**
    - Examine the S3 bucket where CloudTrail logs are stored for any unusual activity, such as attempts to delete or modify logs, which could indicate a compromise.

Categorise the Incident

- **Determine the Cause:**
    - Was CloudTrail disabled intentionally (e.g., maintenance, misconfiguration) or maliciously (e.g., by a compromised account or insider threat)?
- **Assess Impact:**
    - Assess whether the disabling of CloudTrail has impacted your ability to detect and respond to potential security incidents. Determine if there are gaps in log data that could have allowed an attacker to operate undetected.

4. CONTAIN / ERADICATE

Containment Steps

- **Re-enable CloudTrail:**
    - If CloudTrail logging has been disabled, immediately re-enable logging and ensure it is capturing logs in all regions and for all accounts. Configure the trail to log to a secure, encrypted S3 bucket.
- **Investigate the Source:**
    - Identify the source of the disabling request. If an unauthorised user disabled CloudTrail, isolate the affected IAM account or EC2 instance that made the change.
- **Revoke Credentials:**
    - If a compromised IAM user or role disabled CloudTrail, revoke their access immediately by disabling their credentials, resetting passwords or rotating keys.
- **Update IAM Policies:**
    - Review and update IAM policies and roles to ensure that only authorised personnel can modify CloudTrail configuration.

Eradication Steps

- **Terminate Malicious Sessions:**
  - If an attacker disabled CloudTrail, terminate their sessions and perform a thorough investigation to determine if they exfiltrated data or carried out other malicious actions.
- **Review CloudTrail Configuration:**
  - Ensure that CloudTrail logging is configured securely and apply Service Control Policies (SCPs) to restrict actions like disabling logging across accounts.
- **Review IAM Permissions:**
  - Audit IAM policies and roles to ensure that no overly permissive policies allowed for disabling CloudTrail. Enforce least privilege principles.

5. RECOVER

Validate Recovery

- **Confirm Logging is Active:**
  - Verify that CloudTrail is fully re-enabled, logging all API activity and sending logs to an encrypted S3 bucket.
- **Audit CloudTrail Logs:**
  - Once CloudTrail is re-enabled, audit the logs for any gaps in event data and any suspicious activity that may have occurred while CloudTrail was disabled.
- **Check for Compromise:**
  - Verify that no sensitive data was exfiltrated during the downtime of CloudTrail and ensure no ongoing malicious activity is happening.

Restore Configurations

- **Reinforce Monitoring:**
  - Update CloudWatch Alarms and CloudTrail settings to ensure better monitoring of logging status and immediate alerts if logging is disabled again.
- **Review Security Posture:**
  - Reinforce IAM roles, CloudTrail configuration and monitoring practices to ensure that such incidents don't happen again. Consider implementing additional layers of monitoring and alerting for any changes to CloudTrail configuration.

6. LESSONS LEARNT

Post-Incident Review

- **Root Cause Analysis:**
  - Analyse the root cause of why CloudTrail logs were disabled. Was it due to a misconfiguration, a deliberate action or a malicious attack?
- **Impact Assessment:**
  - Assess the impact of the incident, including any gaps in visibility into activity within your AWS environment while CloudTrail was disabled.
- **Update Incident Response Procedures:**
  - Ensure that the incident response procedures are updated to specifically address CloudTrail logging issues, such as having an automated remediation process or additional detection tools.

Enhanced Security Practices

- **Automate Detection and Remediation:**
  - Automate detection and remediation of CloudTrail configuration changes using AWS Lambda or other automation tools to automatically re-enable logging if it is disabled.
- **Increase Security of Configuration Changes:**
  - Apply stricter IAM policies and SCPs to ensure that only authorised personnel can modify CloudTrail settings or disable logging.
- **Continuous Auditing:**
  - Implement a continuous auditing process to monitor for CloudTrail logging issues and regularly review IAM policies to ensure they are aligned with best practices.