AWS Incident Response Playbook:

EC2 Port Scan & Quarantine

1. Detect the Incident

- GuardDuty detects suspicious port scanning activity against EC2.
- Alert is generated (can be real or simulated with AWS sample findings).

2. Investigate

- Use AWS CloudTrail to review events and confirm the alert details.
- Filter for EC2-related events and review actions taken.

3. Contain

- Execute Lambda function `QuarantineInstance`.
- Lambda creates a new “quarantine” security group (no inbound rules) and attaches it to the affected instance, cutting off external access.

4. Document

- Take screenshots of GuardDuty finding, CloudTrail logs, Lambda execution, and EC2 security group status.
- Save logs and findings for evidence.

5. Report

- Write an incident timeline summarizing what happened and actions taken.

Incident Timeline

Time	Action
19:25	Simulated GuardDuty finding via AWS CLI.
19:28	Verified GuardDuty finding in AWS Console.
19:29	Reviewed CloudTrail event for CreateSampleFindings.
19:35	Created Lambda function for EC2 quarantine.
19:38	Attached permissions to Lambda function.
19:44	Lambda succeeded; EC2 instance quarantined.
19:46	Confirmed quarantine security group attached.
19:47	Verified quarantine SG has no inbound rules.
19:48	Saved all screenshots for documentation.

Screenshots & Deliverables

Action	Screenshot Name	Description
Enable GuardDuty	GuardDuty-Enabled.png	GuardDuty enabled in AWS Console.
Create CloudTrail Trail	CouldTrail-TrailCreated.png	CloudTrail multi-region trail created.
Review CloudTrail Events	CloudTrail-EventHistory.png	CloudTrail event history review.
Launch EC2 Instance	EC2-Instance-Running.png	EC2 test instance running.
Set Security Group Open to SSH	EC2-SecurityGroup-SSH-Open-png	Security group with SSH open.
Simulate GuardDuty Finding	GuardDuty-Simultated-PortProbe-Finding.png	Simulated GuardDuty finding.
Create Lambda Function	Lambda-Code.png	Lambda quarantine function code.
Assign Lambda Permissions	Lambda-Permissions.png	Lambda execution role permissions.
Run Lambda & Verify Success	Lambda-Quarantine-TestSuccess.png	Lambda success output: instance quarantined.
Confirm EC2 Quarantined	EC2-Quarantined-SecurityGroup-png	EC2 is attached to the quarantine security group.
Verify No Inbound Rules in Quarantine SG	QuarantineSG-NoInboundRules.png	Quarantine security group: no inbound rules

Reflection

What I Learned

During this lab, I learned how AWS's cloud-native security tools work together to detect, investigate, and automatically respond to security incidents. I gained hands-on experience configuring GuardDuty, reviewing detailed audit logs with CloudTrail, and building a Lambda function to isolate compromised EC2 instances using security group automation. I also learned how to document each step with evidence—an essential skill for real security teams.

What I'd Improve Next Time

If I were to do this lab again, I'd work on triggering a real GuardDuty alert using different types of activity (like brute-force SSH attempts or multiple port scans from various sources) to see more diverse findings. I would also look into improving the Lambda code to automatically get the instance ID from the actual GuardDuty finding instead of hard-coding it. Additionally, I would script or automate more of the documentation process to save time and reduce manual steps.

How I'd Automate More Steps

For future improvement, I would connect GuardDuty alerts directly to Lambda using EventBridge rules, so incidents are detected and contained automatically without manual intervention. I'd also set up notifications (like email or Slack) to alert the security team whenever the Lambda function is triggered, and consider adding automated evidence collection and backup before quarantine actions are performed.