Sprint 4

Creating a GuardDuty Threat List and Integrating with Security Hub

This guide outlines the steps to create a GuardDuty threat list, test it with EC2 instances, and integrate the findings into Security Hub for improved security visibility:

Step 1: Create a GuardDuty Threat List

1. **Prepare the Threat List:**
    - Create a CSV file named `threatlist.csv` containing the IP addresses you want to monitor.
    - Example:

```
192.0.2.0/24
198.51.100.0/24
```

2. **Upload the Threat List to an S3 Bucket:**
    - Upload the `threatlist.csv` file to an S3 bucket in your AWS account.
3. **Enable GuardDuty and Create a Threat List:**
    - If not already enabled, navigate to the AWS Management Console and enable GuardDuty.
    - In the GuardDuty console, under "Lists," create a new Threat List.

- Provide the S3 URI of the uploaded CSV file when creating the Threat List.

Step 2: Create and Configure EC2 Instances

1. **Launch Two EC2 Instances:**
    - Launch two EC2 instances in your AWS account, ensuring they are in the same region where GuardDuty is enabled.
2. **Simulate Malicious Activity:**
    - Connect to one of the EC2 instances.
    - Simulate traffic to one of the IP addresses in your threat list. You can use `curl` to make an HTTP request:
    - Bash

```
curl http://192.0.2.1
```

    -
    - (Replace `192.0.2.1` with the actual IP address from your list)

Step 3: Monitor GuardDuty Findings

1. **Check GuardDuty Findings:**
    - After simulating activity, navigate to the GuardDuty console.

- Under "Findings," check for new findings related to the simulated activity. GuardDuty should detect connections to the IP addresses in your threat list.

Step 4: Integrate GuardDuty Findings into Security Hub

1. **Enable Security Hub:**
    - If not already enabled, navigate to the Security Hub console and enable Security Hub.
2. **Enable GuardDuty Integration:**
    - In the Security Hub console, go to the "Integrations" section.
    - Enable the GuardDuty integration if it's not already enabled.
3. **View Findings in Security Hub:**
    - After enabling the integration, navigate to the "Findings" section in Security Hub.
    - You should see GuardDuty findings listed along with other security findings.

Step 5: Automate Responses (Optional)

This step outlines an optional approach to automate responses to GuardDuty findings:

1. **Set Up Automated Responses with AWS Lambda:**
    - You can configure automated responses using AWS Lambda and CloudWatch Events.

2. **Create a CloudWatch Event Rule:**

    - Go to the CloudWatch console and create a new rule with the following event pattern to match GuardDuty findings:

    - JSON

```
{
  "source": ["aws.guardduty"],
  "detail-type": ["GuardDuty Finding"]
}
```

    -
    -
    - Set the target of the rule to the Lambda function that you want to invoke when a GuardDuty finding triggers the event.

3. **Create a Lambda Function:**

    - Create a Lambda function to handle GuardDuty findings. The function can take actions like stopping an instance or sending notifications based on the finding details.

```
import boto3

def lambda_handler(event, context):
```

```
    ec2 = boto3.client('ec2')
    guardduty = boto3.client('guardduty')
    finding = event['detail']
    instance_id =
finding['resource']['instanceDetails']['instanceId']
    try:
        response = ec2.stop_instances(InstanceIds=[instance_id])
        print(f"Stopped instance: {instance_id}")
    except Exception as e:
        print(f"Error stopping instance: {str(e)}")
```

Important Notes:

- Ensure the Security Hub integration with GuardDuty is enabled for findings to appear.
- Consider testing the Lambda function independently before integrating it with GuardDuty findings to ensure proper functionality.
- Remember to follow AWS best practices for security and access control when setting up these services.

By following these steps, you can create a GuardDuty threat list, monitor suspicious activity, and integrate the findings with Security Hub for a centralized view of your

security posture. Optionally, you can further enhance your security by automating responses using AWS Lambda and CloudWatch Events.

Technical Documentation (Focused on SIEM with Limited SOAR Integration)

SOAR Tool Selection and Integration (Future Consideration)

- **Chosen Tool (Placeholder):** While a full SOAR suite isn't implemented here, consider integrating a SOAR platform like Palo Alto Networks Cortex XSOAR or MacAfee MVISION SOAR in the future. These platforms offer comprehensive SOAR features and integrate with various AWS services.
- **Integration Process (Future Consideration):** Integration with a chosen SOAR platform would involve establishing secure communication channels between the SOAR tool and your AWS services. This might involve API integrations or dedicated connectors provided by the SOAR vendor. Security Hub could play a role here, as it aggregates security findings from AWS services and can be integrated with some SOAR platforms.
- **Challenges (Future Consideration):** Potential challenges include managing API keys and access controls for secure communication, ensuring data format compatibility between platforms, and potential limitations in the free tiers of some SOAR solutions.

Incident Response Playbooks (Limited with SIEM)

- **Overview:** While a full SOAR suite isn't implemented here, we can leverage basic automation within the SIEM solution for some incident response tasks. Playbooks might be limited in scope but can still be beneficial.
- **Incident Types:** Playbooks could address common security incidents like suspicious login attempts detected by CloudTrail, potential data exfiltration identified through VPC Flow Logs, or vulnerabilities discovered by Inspector.
- **Workflow and Automation:** Playbooks could trigger automated actions like user account lockout for suspicious logins, network traffic filtering for potential exfiltration attempts, or automated isolation of vulnerable instances.
- **Security Best Practices:** Playbooks should adhere to best practices like NIST Cybersecurity Framework or MITRE ATT&CK. Leverage these frameworks to define response procedures based on the identified incident type.

Playbook Testing and Validation

- **Testing and Validation:** Develop test scenarios simulating security incidents. These scenarios could involve injecting test data into CloudTrail logs or VPC Flow Logs to trigger playbooks and validate their functionality.
- **Security of Playbooks:** Implement access controls and permissions within the SIEM solution to restrict unauthorized modification of playbooks. Regularly review and update playbooks to ensure they remain effective and don't introduce vulnerabilities.

Threat Detection & Alert Correlation

SIEM Solution and Data Sources

- **SIEM Solution:** We'll leverage Amazon CloudWatch Logs as the central log management solution. CloudTrail, VPC Flow Logs, and Inspector findings can all be integrated with CloudWatch Logs for centralized analysis.
- **Data Sources:** Data sources include CloudTrail logs (API activity), VPC Flow Logs (network traffic), and Inspector findings (vulnerability scans).
- **Normalization and Enrichment:** CloudWatch Logs offers log parsing capabilities to normalize logs from different sources into a common format. Additional enrichment might involve integrating threat intelligence feeds to add context to log data.

Correlation Rules and Threat Detection

- **Correlation Rules:** Develop rules within CloudWatch Logs that analyze log data from various sources to identify potential threats. These rules might look for specific patterns in CloudTrail logs indicating suspicious activity or correlate network traffic logs with potential vulnerabilities identified by Inspector.
- **Techniques:** Techniques could include pattern matching (e.g., identifying known malicious commands in CloudTrail logs), statistical analysis (e.g., detecting unusual spikes in network traffic), or simple rule-based correlations.
- **Threat Intelligence and IoCs:** Integrate threat intelligence feeds from reputable providers to enrich correlation rules with the latest indicators of compromise (IoCs). This helps identify emerging threats and suspicious activity patterns.
- **Tuning and Refinement:** Continuously monitor and refine correlation rules to reduce false positives and improve detection accuracy. Analyze false positives to understand the root cause and adjust rules accordingly.

Alert Prioritization and Triage

- **Prioritization:** Prioritize alerts based on a combination of factors like asset criticality (e.g., high-risk servers), threat level (e.g., potential data breach), and confidence score (e.g., high confidence based on multiple indicators).
- **Integration with Playbooks:** Alerts triggered by correlation rules can be routed to basic automation within the SIEM solution or integrated with a future SOAR platform for more complex workflows.
- **High-Fidelity Alerts (Example):** A high-fidelity alert might be triggered if CloudTrail logs show a successful login attempt from an unauthorized IP address for a critical server account. This could initiate automated playbook actions to lock the compromised account and notify security personnel for further investigation.