**Control Description (describe the control using 5W2H)**

Container security scanning is implemented to actively monitor and validate critical security configurations such as RBAC, network policies, security settings, and resource limits (what). This scanning is applied at the node, cluster, pod, and container levels in both Kubernetes and ECS environments (where) to ensure compliance and prevent unauthorized modifications (how). It is performed continuously throughout the deployment and runtime processes (when), managed by authorized security teams and automated tools (who), ensuring the integrity and security of containerized environments (how much).

**Control Objective (why is the control performed and how does it manage an issue?)**

The control ensures compliance with security standards, prevents misconfigurations, and detects unauthorized changes. By continuously monitoring critical configurations (e.g., RBAC, network policies, and resource limits), the control mitigates risks of privilege escalation, data breaches, and operational instability in containerized environments.

**Control Frequency (how often is the control performed?)**

The control is performed **continuously** during both the deployment and runtime phases.

**Control Ownership (who performs the control and who is responsible for the control?)**

* **Performed by:** Authorized security teams and automated tools.
* **Responsible for the control:** Security and DevOps teams.

**Control Type (e.g., preventative/ detective)**

The control is **preventative** and **detective** in nature as it prevents unauthorized changes and detects misconfigurations.

**Control Classification (e.g., manual/ Automated)**

The control is **automated** with oversight by security teams.

**Where is it performed (e.g., location, system)?**

The control is performed across the **node, cluster, pod, and container levels** in both **Kubernetes** and **ECS environments**.

**Background Detail (if required)**

Containerized environments are critical to modern deployments, and any misconfigurations in RBAC, network policies, or resource limits can lead to security breaches or unauthorized access. This control ensures real-time validation and enforcement of security policies to maintain compliance with industry standards.

**How is the control performed (describe the control key steps on a high-level)?**

* **Inputs and Sources:**
  + Configuration files (RBAC policies, network rules, resource limits)
  + Security tools and scanners integrated into the container orchestration platforms (e.g., Kubernetes, ECS)
  + Deployment pipelines and runtime data logs
* **What is done to process the inputs:**
  + Automated scanning tools validate security configurations against predefined security baselines and CIS benchmarks.
  + Configurations (e.g., RBAC permissions, network segmentation policies, resource limits) are continuously monitored for compliance.
* **Relevant thresholds/limits:**
  + Misconfigured permissions (e.g., overly permissive RBAC roles)
  + Non-compliance with network isolation policies
  + Resource limits not enforced as per defined baselines
* **How outliers are tracked, managed, escalated:**
  + Non-compliant or misconfigured components are flagged by the scanning tools.
  + Alerts are generated and escalated to the security and DevOps teams for immediate remediation.
  + Non-compliance trends are reported for management review.
* **How the control review is evidenced:**
  + Automated scan reports from security tools
  + Logs showing flagged misconfigurations, escalations, and remediation actions
  + Periodic compliance review reports generated by authorized security teams.

**Control: Container Security Configuration Scanning and Reporting**

**Control Test: Validation of Security Configuration Scanning and Reporting**

**Control Test:** Verify that critical security configurations of containers are being scanned and compliance/non-compliance is reported to senior management/CIOs.

**Control Attribute:**

* Critical configurations of the containers are being scanned.
* Reporting of compliant and non-compliant security configurations to higher management/CIOs.

**Test Evaluation Criteria:**

* **Defined by:** CSAM (Cloud Security Assurance Methodology) policies approved by Security Architecture (SecArch) for EKS, ECS, and AKS services.
* **Reasonability:** The control must ensure that scanning tools actively monitor configurations, detect changes, and report non-compliant configurations to senior management. The expectation is aligned with CSAM standards.

**Key Test Steps:**

1. **Sub-sample Selected and Details:**
   * Selected CSAM policies for Amazon EKS, ECS, and Azure AKS services to verify the implementation of a detective control for container configuration changes.
2. **What Was Looked At:**
   * **CSAM Policies:**
     + ECS: Amazon ECS - CSAM Framework - Confluence Enterprise.
     + EKS: Amazon EKS - CSAM Framework - Confluence Enterprise.
     + AKS: Azure AKS - CSAM Framework - Confluence Enterprise.
   * Cloud Security Team discussions to validate the current state of configuration scanning.
3. **How the Test Was Executed:**
   * **For ECS Services:**
     + Audit confirmed with the Cloud Security Team via call that no scanning policies or tools are being developed for ECS as migration to EKS (NEF2) is underway.
   * **For EKS Services:**
     + Audit confirmed that the team is in the process of implementing a new tool to scan critical security configurations outlined in CSAM policies.
   * **For AKS Services:**
     + Audit confirmed that the team is also bringing a new tool for AKS to scan configurations in line with CSAM policies and report findings to senior management.

**Outcome:**

* **Result/Condition:** There is no security configuration scanning currently in place for ECS, EKS, or AKS services.
* **Comparison Against Expectations:**
  + **Expected:** Scanning tools should actively validate configurations and detect deviations, with findings reported to senior management.
  + **Actual:** No tools or processes are operational for scanning or reporting security configurations for ECS, EKS, or AKS.

**Impact Statement:** The lack of security configuration scanning poses a significant risk to the integrity and security of containerized environments. Without proper scanning and reporting mechanisms:

* Misconfigurations in critical areas such as RBAC, network policies, and resource limits may go undetected, increasing exposure to unauthorized access, privilege escalation, and operational instability.
* Non-compliance with NAB security policies and industry best practices (e.g., CIS benchmarks) could result in regulatory and reputational risks.
* Senior management lacks visibility into security posture, hindering timely mitigation of vulnerabilities.

**Reason for Control Failure:** The control failed due to the following reasons:

1. **Absence of Security Tools:** There are currently no scanning tools implemented for ECS, EKS, or AKS environments to monitor critical configurations.
2. **Incomplete Implementation Plans:** While tools for EKS and AKS are in progress, they are not yet operational, leaving a gap in detecting configuration deviations.
3. **Lack of Preventive or Detective Mechanisms for ECS:** The Cloud Security Team confirmed that ECS scanning policies are not being developed due to the ongoing migration to EKS, leading to unmanaged security risks in the interim period.
4. **Reporting Gap:** No mechanism is in place to report compliance or non-compliance to senior management, limiting visibility and timely action on misconfigurations.

**Rationale for Pass/Fail:**

* The control **failed** because no scanning tools are implemented to monitor critical container security configurations, and there is no mechanism to report compliance/non-compliance to senior management.

**Control Test Result:**

* **Fail:** The control did not ensure security configuration scanning and reporting for ECS, EKS, and AKS environments.

**Exceptions:**

* **Exception 1:** ECS services have no scanning policies in place as migration to EKS is ongoing.
* **Exception 2:** EKS and AKS services do not yet have tools implemented to perform security configuration scans or report findings to senior management.

**Recommendations:**

1. Implement security scanning tools for ECS, EKS, and AKS services to monitor and validate critical configurations.
2. Establish a process to report compliance/non-compliance to senior management as per CSAM requirements.
3. Prioritize the migration of ECS to EKS with immediate interim controls for ECS security monitoring.

**Control Assessment: Design Effectiveness**

| **Question** | **Result** | **Rationale** |
| --- | --- | --- |
| **Is this the right control to mitigate the risk?** | **Fail** | The control aims to monitor critical container security configurations, but it is ineffective due to the absence of scanning tools for ECS, EKS, and AKS environments. **(Exception 1, 2)**. |
| **Does the control achieve its objective?** | **Fail** | The control fails to achieve its objective as there are no operational tools to validate or report on critical configurations. **(Exception 1, 2)**. |
| **Is the control performed by the right people with requisite skills?** | **Pass** | The Cloud Security Team and Security Architect possess the required skills and knowledge, but tools and processes are yet to be implemented. |
| **Does the control have adequate segregation of duties?** | **Pass** | The segregation of duties is in place, as the Cloud Security Team manages the scanning process independently of development teams. |
| **Is the control performed at the right time or stage of the process?** | **Fail** | The control is not being performed at any stage, as scanning tools for EKS, ECS, and AKS are either absent or in progress. **(Exception 1, 2)**. |
| **Is the control performed at the right frequency?** | **Fail** | The control is not performed continuously as required, due to the absence of scanning tools. **(Exception 1, 2)**. |
| **Is the control sustainable?** | **Fail** | The current approach is unsustainable as no interim controls are in place for ECS and tools for EKS/AKS are incomplete. **(Exception 1)**. |
| **How does it manage/escalate an issue?** | **Fail** | There is no mechanism to report non-compliant configurations to senior management, limiting timely issue escalation. **(Exception 2)**. |
| **Is the control evidenced?** | **Fail** | There is no evidence of security configuration scanning or reporting, as the tools are not operational. **(Exception 1, 2)**. |
| **Does the control have adequate management focus?** | **Fail** | Management focus on implementing scanning tools for ECS, EKS, and AKS is insufficient, leaving gaps in security governance. **(Exception 1, 2)**. |

**Overall Summary:**

The control is ineffective as security configuration scanning tools are not implemented for ECS, EKS, and AKS environments. This creates a significant risk of misconfigurations going undetected and unreported, exposing containerized environments to potential threats such as unauthorized access, privilege escalation, and operational instability.

**Exceptions:**

* **Exception 1:** ECS lacks security scanning policies and tools, with no interim measures during migration to EKS.
* **Exception 2:** Tools for EKS and AKS are still being implemented and are not operational, with no reporting to senior management.

**Recommendation:**

Immediate action is required to:

1. Implement interim security controls for ECS.
2. Expedite the deployment of scanning tools for EKS and AKS environments.
3. Establish a mechanism for reporting compliance and non-compliance to senior management.