iterasec

Securing NDA Security company's Multi-Tenant Architecture

Client:
NDA Security company

Industry:
Network Security and Infrastructure

Company Size:
300+ Employees

Services Used:
Cloud & Container Security Testing, Kubernetes Audit, AWS Security Review

Client

An NDA Security company specialized in advanced solutions for network security, providing both software and hardware products designed to protect complex enterprise environments. Their technology empowered organizations to manage threats and vulnerabilities effectively, even in highly distributed and rapidly changing infrastructures. Through continuous innovation and a technically skilled workforce, company served a variety of sectors that demanded robust, adaptable solutions.

Background

Building on the success of their single-tenant offering, a Security company introduced a multi-tenant version of their core product. While functionally similar at the application layer, this new version shared certain infrastructure components among different clients. With a goal of efficiently scaling their services while maintaining strict isolation of each client's data, a Security company once again partnered with Iterasec to ensure that multi-tenant deployments met the highest security standards.

The Challenge

- **Shared Infrastructure Complexity**
  A Security company's multi-tenant architecture required critical shared resources such as container orchestration, cloud environments, and multiple 3rd party applications, to remain secure and properly segmented among various clients.
- **Kubernetes & AWS Security**
  - The move to container-based deployments presented unique challenges, particularly around Kubernetes cluster configuration, container security, and cloud integrations.
  - Existing AWS services had to be configured to prevent cross-tenant access, calling for an in-depth review of identity and access management (IAM) controls, storage configurations, and network setups.
- **Strict Isolation Requirements**
  A primary objective was to verify that each tenant's environment was fully isolated, ensuring no client could access or interfere with another client's resources.

The Solution

Iterasec conducted a specialized security assessment focused on containers and cloud infrastructure, using both automated tools and in-depth manual techniques:

Kubernetes Audit

- **Analyzed cluster configurations, validating role-based access controls (RBAC) and runtime policies.**
- **Assessed the security of the container registry and running containers, focusing on misconfigured access controls and image build best practices.**

AWS Security Review

- **Evaluated identity and access management settings, checking for overly permissive policies.**
- **Reviewed logging and monitoring solutions to ensure effective threat detection and incident response capabilities.**
- **Assessed the security of IAM Roles for Service Accounts, checking for misconfigured role trust policies.**
- **Investigated network segmentation strategies, storage configurations, and secret management settings to confirm data isolation and integrity.**

Multi-Tenant Testing Scenarios

- **Evaluated identity and access management settings, checking for overly permissive policies.**
- **Reviewed logging and monitoring solutions to ensure effective threat detection and incident response capabilities.**
- **Assessed the security of IAM Roles for Service Accounts, checking for misconfigured role trust policies.**
- **Investigated network segmentation strategies, storage configurations, and secret management settings to confirm data isolation and integrity.**

The Outcome

- **Validated Multi-Tenant Isolation**
  - Verified that the multi-tenant architecture effectively prevented cross-client data exposure, reinforcing confidence in company's product.
  - Identified severe vulnerabilities in both cross-tenant data isolation and tenants' secret management, finding "Unrestricted Access to IMDS", "EC2 LaunchTemplates do not Require IMDSv2".
  - Provided the client with a detailed roadmap on improving the secret management process.
  - Offered significant improvements of AWS EFS access policies to prevent unwanted access to Kubernetes Persistent Volumes, finding "Improper Access Control on EFS Shares".
  - Proposed improvements for data encryption both in transit and at rest, as well as improvements in KMS policies for more robust cross-tenant data isolation, finding "Identical KMS Permissions in Different Tenants", "Improper TLS Configuration in S3 Buckets", "Insecure Transportation Security Protocols Supported".
- **Enhanced Container Security**
  - Identified and addressed specific vulnerabilities in container builds, such as hardcoded credentials, violations of the principle of least privilege, and dependency on vulnerable 3rd-party components — findings "Improper Credential Storage", "Use of Outdated Components in Containers".
  - Offered several strategies to prevent unbounded resource consumption in Kubernetes-managed containers, improving the overall resilience of the platform ("Lack of Resource Consumption Limits in Kubernetes Containers").
  - Helped implement a Kyverno Policy-as-Code solution to mitigate the deployment of overly privileged containers ("Kubernetes Node Escape via Privileged Pods").
  - Raised awareness about potential threats arising from misuse of privileged 3rd-party monitoring solutions ("Kubernetes Containers Lateral Movement via Misconfigured Mounts").
- **Strengthened AWS Posture**
  - Implemented tighter IAM controls and refined network policies, reducing the risk of misconfigurations in a shared cloud environment.
  - Offered enhancements for EC2 Launch Templates configurations that restrict unwanted access to cross-tenant data, configuration of the instance, and improve resilience to Server-Side Request Forgery attacks — finding "EC2 LaunchTemplates do not Require IMDSv2".
  - Unveiled hidden risks stemming from overly permissive IAM Role Trust Policies for principals in external AWS accounts.
- **Roadmap for Continuous Improvement**
  - Provided recommendations on container lifecycle management, automated patching, and ongoing security monitoring to support the rapid evolution of a Security company's multi-tenant services.
- **Addressing Amplified Single-Tenant Vulnerabilities**
  - Demonstrated how some weaknesses identified in the single-tenant version — such as SQL injection — could pose an even greater threat in a containerized environment, with more components than they did in a single-tenant version of the software.
  - Provided additional safeguards and best practices to ensure that a compromise in one tenant does not cascade and jeopardize the broader shared infrastructure, emphasizing critical Kubernetes and AWS configuration checks.

Conclusion

Through a targeted assessment of Kubernetes, AWS, and the underlying multi-tenant architecture, Iterasec helped a Security company ensure that multiple clients could share infrastructure components without compromising data confidentiality and integrity. By uncovering and resolving potential risks in container deployments and cloud configurations, the partnership reinforced company's commitment to delivering scalable, secure solutions that meet the stringent demands of modern enterprise environments.