

Adopting Zero Trust Architecture for Robust Supply Chain Protection

MSc Research Project
Masters in Cybersecurity

Harsh Nailesh Doshi
Student ID: X22207848

School of Computing
National College of Ireland

Supervisor: Eugene Mclaughlin

National College of Ireland
MSc Project Submission Sheet
School of Computing



Student Name: Harsh Nailesh Doshi
Student ID: X22207848
Programme: Masters in Cybersecurity **Year:** 2024-25
Module: Practicum Part 2
Supervisor: Eugene McLaughlin
Submission Due Date: 12-12-2024
Project Title: Adopting Zero Trust Architecture for Robust Supply Chain Protection

Word Count: 5521 Page Count: 19

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:

A handwritten signature in blue ink, appearing to be "Harsh Doshi", written over a light blue grid background.

Date: 12-12-2024

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	<input type="checkbox"/>
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	<input type="checkbox"/>
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	<input type="checkbox"/>

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Adopting Zero Trust Architecture for Robust Supply Chain Protection

Harsh Nailesh Doshi
X22207848

Abstract

The present supply chain has ill-defined and porous perimeter, making it vulnerable to intercept sensitive information and disrupt operations. This research investigates the introduction of ZTA into supply chain cybersecurity, with particular emphasis on micro-segmentation, secure identity management, and real-time monitoring that exploit tools such as Istio, SPIRE, and HashiCorp Vault. To achieve this, a Kubernetes-based environment was simulated to assess ZTA's effect on risk-minimization and resilience-enhancement. ZTA principles, although showing remarkable security improvements, also brought forward a lot of impediments. Just to name a few, the integration of complex tools, the solution's scalability, and limited monitoring were the highlighted challenges. This study will explore the practical steps for needed for ZTA implementation, challenges faced, and the impact of operational efficiency. Key findings will contribute to developing a framework to integrate ZTA in supply chain management.

Keywords: Zero Trust Architecture, Supply Chain Security, Continuous Verification, Operational Efficiency, Policy Enforcement, Kubernetes, Istio, SPIRE, HashiCorp Vault, Real-Time Monitoring, Secure Communication.

1 Introduction

The rise in interconnectedness and complex nature of supply chain makes it vulnerable to cyber threats that are sophisticated. The Zero Trust Architecture (ZTA) was launched by an analyst of Forrester Research in 2010 keeping internet security in mind. Most traditional models rely on perimeter defences which is becoming inadequate with time in protecting cyberattacks on complex supply chains. ZTA offers a more robust security approach by assuming that the threats can come both within and outside the network by continuously verifying and validating every access request to Policy Decision Point (PDP) and Policy Enforcement Point (PEP). This approach is especially relevant given the recent high-profile breaches, such as the SolarWinds attack, which highlights the need for more stringent measures (Rose, S et al. 2020).

ZTA can be implemented in different domains as well, such as finance where ZTA can conduct real-time reviews of the business's amount flow, tax payment ability, repayment willingness, and business capability of small and medium-sized enterprises based on communication protocols whilst obtaining real-time data (Pan, Y et al. 2023). ZTA can utilise its intelligent identity verification to ensure that the financial business data of the enterprise is

only getting shared within the same supply chain, isolating and distinguishing enterprises of competitive relations. This involves designing an access control models in ZTA.

Recent studies that are already available are mostly about securing one supply chain nodes or processes. Nonetheless, the truth is that the mutual dependency of stakeholders calls for a more all-encompassing direction. The introduction of ZTA principles into supply chains will solve the issues by restricting access, utilizing continuous monitoring, and data encryption in all endpoints. Despite its potential, implementing ZTA in supply chains introduces challenges related to scalability, compatibility, and cost (Gupta & Bharathi, 2023; Van Bossuyt et al., 2023).

1.1 Research Question

How can Zero Trust Architecture enhance cybersecurity in supply chains, and what are the practical steps for its implementation?

ZTA works under the impression that the threats can originate from both outside and within the network, warranting a switch from traditional perimeter-based defences to a model which requires strict verification for every user and device attempting to access resources. By deploying concrete initiatives like micro segmentation with various tools such as Istio, identity management with SPIFFE/SPIRE, and secrets management using HashiCorp Vault, ZTA thus takes care of the interconnected and vulnerable nature of supply chains. Continuous review and control of policies through tools like Open Policy Agent in turn further secure the security. This is a problem that must be dealt with because of the increasing frequency of cyberattacks, the operational importance of supply chains, and the vulnerabilities posed by third-party collections. ZTA not only keeps the supply chains solid against such challenges but also makes sure that the system is compliant with the requirements and reduces the expenses of the damages and disruptions over time.

1.2 Objective

To address the research question and ensure the relevance of the proposed implementation framework, this study focuses on the following objectives:

1. Design a Zero Trust framework for supply chains based on modular, scalable technologies and robust access controls.
2. Define actionable steps for deploying ZTA, including micro segmentation, secure identity management, and real-time monitoring.
3. Validate the framework through simulations and practical demonstrations to ensure its effectiveness in real-world scenarios.

2 Related Work

In Gupta, V et al. (2023) paper explore the application of ZTA in managing cybersecurity risks within supply chains, emphasizing its transformative potential in addressing vulnerabilities inherent to modern, interconnected systems. The study contextualizes its research within the limitations of traditional perimeter-based security models, highlighting ZTA's principles such as continuous verification, least-privilege access, and micro

segmentation as proactive measures to enhance supply chain security. By outlining practical implementation steps, including identity management, secure communication, and access controls, the research bridges the gap between theoretical constructs and actionable solutions. Its focus on scalability and adaptability across diverse supply chain contexts further enhances its relevance. While the paper's framework is robust and innovative, it identifies challenges such as the lack of empirical validation and limited exploration of economic feasibility for small and medium enterprises (SMEs). The study assumes uniform collaboration among supply chain stakeholders, leaving potential issues of coordination and legacy system integration underexplored. Nonetheless, the research substantiates the importance of ZTA in fortifying supply chains against cyber threats and provides a solid foundation for future studies. Addressing gaps such as sector-specific customization and validating the framework through case studies or simulations could further strengthen its practical applicability. This work aligns with the research question by demonstrating how ZTA can enhance supply chain resilience and offering actionable insights for implementation.

Van Bossuyt, D.L et al. (2023) does the overall analysing of the integration of ZTA into the whole cycle of system design, acknowledging the increasing embracement of security due to the threats of AI/ML vulnerabilities, data poisoning, and adversarial attacks. By broadening the ZTA standards beyond the only IT setups, the authors develop a framework that secures all the lifecycle stages from the beginning of the project to its end (including the disposal), in areas like supply chain vetting, digital twin security, and the development of the secure system inputs. The papers' expanded scope and the given practical suggestions, such as concurrent firewalled design teams and continuous monitoring, form an adequate foundation for the development of the cyber-physical system that easily withstands any setbacks. In the study ZTA's flexibility in changing situations and following new trends like model-based systems engineering (MBSE) and digital twins are of importance. The framework is solid in theory, absence of data validation, scalability considerations, and detailed cost analyses inhibit its implementation practice. The study assumes that there is a universal ZTA model, thus, it ignores the differences in various industries and the harsh problems that are associated with legacy system integration. However, its focus on a lifecycle perspective truly meets the needs of the existing literature, which is very often geared towards ZTA as a standalone issue. By emphasizing the future investigation of sector-oriented modifications, scalability, and less-cost solutions, the paper significantly facilitates the secure, flexible, and strong systems' development made possible by ZTA. It supports the fact that ZTA should be used as the basis for managing security risks in the modern engineering sectors.

Amaral, T & Gondim, J (2021) explores the integration of ZTA into cyber supply chains as a proactive approach to mitigate growing security vulnerabilities. Using frameworks such as NIST SP 800-207, it organizes security controls across six domains, including infrastructure, identity, governance, and DevSecOps. Key tools, such as the Software Bill of Materials (SBOM), are emphasized for tracking software dependencies and identifying potential risks. The study also provides a structured checklist and visualization techniques to facilitate gap analyses and design security improvement roadmaps. By adopting dynamic policies, real-time monitoring, and automated controls, the proposed framework aims to address critical issues like access control, configuration management, and incident

response. While the framework is comprehensive, the paper lacks empirical validation and detailed cost analysis, which are crucial for assessing the feasibility of implementation. The generality of the proposed controls may limit their effectiveness across diverse sectors, and challenges such as legacy system integration and stakeholder collaboration are not adequately addressed. Despite these gaps, the research significantly advances the discourse on supply chain cybersecurity by operationalizing ZTA principles and providing actionable tools for risk management. Future research focused on sector-specific adaptations, cost-effective solutions, and real-world testing will be essential to validate and refine the proposed framework.

Collier, J & Sarkis, Z (2021) conceives a vision of a "Zero Trust Supply Chain," using ZTA principles as a start to handle risks in modern networked supply chains. It outlines the insufficiency of traditional trust-centred approaches in handling vulnerabilities brought by a more complex and continuously evolving international supply chain. By utilizing key ZTA principles such as continuous verification, least privilege, and logging, the authors recommend a holistic framework for controlling risks and minimizing damage. The paper is based on concepts from the organizational theories such as institutional theory and transaction cost economics and at the same time connects the Zero Trust phenomenon with these two dimensions, so it is shown to be a systemic approach instead of a simple list of measures, moreover, it provides the necessary practical steps for the emergence of a Zero Trust Supply Chain. While the framework is conceptually sound and theoretically innovative, the paper suffers from a lack of empirical validation and does not address the operational trade-off issues and the economic feasibility of implementations in different supply chain settings. Besides that, its solution is general and doesn't address the needs of different sectors or the potential conflicts that might appear between the Zero Trust methods and the fact that such systems are mainly collaborative. However, the research findings have potential implications on the supply chain security front, as it augments the track of the discourse by offering a solid theoretical base for zero trust implementation which makes it an inevitable strategic priority for the organizations that must cope with the multiplying cyber and operational risks. Further studies are needed to investigate what the practical ramifications of a zero-trust approach are and whether companies should still invest in cybersecurity when such an approach is taken.

Pan, Y et al. (2023) presents a supply chain finance (SCF) model that builds in ZTA and blockchain technologies to solve the perpetual problems that small and medium-sized enterprises (SMEs) are facing, such as long credit evaluations and inflexible approval processes. The scheme takes advantage of ZTA tenets, such as continuous verification and least privilege, as well as blockchain's distributed and secure data management, to make a responsive and secure financing network. The major characteristics of these include online credit checking, smart contracts for the automated decision-making process, and decentralized identity management which ensures secure communication across the supply chain. The multi-levelled architecture which consists of preprocessing, evaluation, and application layers process the credit transactions thus, it improves the quality of financing and prioritizes the needs of SMEs which are typically the underprivileged in the existing SCF models. While the framework provides a theoretically solid and novel solution, it is confronted with problems in practical implementation. Problematically, no empirical proof

has been provided, for instance, in the form of case studies or simulations which leaves space for only a low degree of reality manifested. Moreover, the technical and financial barriers that SMEs will face in the deployment of blockchain and ZTA technologies as well as the challenges of collaborative interaction and the flexibility of each sector are still to be thoroughly analysed. Despite these gaps, the research showcases the potential of ZTA and blockchain to revolutionize SCF by ensuring high transparency, security, and adaptability. It makes a foundation for the next study to give knowledge of experimentation and economic viability, thereby it bought a larger number of actual cases with efficiency.

Paul, B et al. (2022) examines how ZTA is applied in manufacturing systems to secure them in Industry 4.0, where decentralized interconnected operations are susceptible to elaborate cyber threats. Legacy perimeter-based security systems, which assume insider networks are trusted by default, fail to meet these threats. The proposed ZTA framework introduces essential concepts such as continuous verification, least privilege access, and micro-segmentation to improve the cybersecurity posture of operating technology environments. Using identity and access management (IAM), secure communication protocols, endpoint compliance monitoring, and firewalls, it provides ways to rectify shortcomings such as weak communication channels and unauthorized access in cyber-physical systems. While the framework is extensive and suitable for industry-based problems in Industry 4.0, it has implementation issues. The absence of empirical evidence from case studies or pilot projects limits its immediate applicability, and the stringent demands of technological expertise and infrastructure may be a stumbling block for smaller manufacturers. Moreover, the paper does not provide full solutions to the issues of economic feasibility, sector-specific modifications, or strategies for integrating ZTA with legacy systems. Nevertheless, the research makes a major contribution to the OT security discussion by introducing a sound theoretical framework for ZTA in smart manufacturing. In the future, these issues should be solved through real-world validation, cost-effective implementations, and scalable solutions to facilitate a widespread implementation.

Subsection 2

3 Research Methodology

The research methodology outlines the structured approach undertaken to study, design, implement, and evaluate the adoption of ZTA in supply chains. The methodological approach is to use ZTA principles in a simulated supply chain environment, which include identity management, micro segmentation, and secure communication protocols. This framework was the result of further development using case-specific scenarios that depict real-world operations in supply chain and manufacturing sectors, thus, guaranteeing the applicability and relevance.

3.1 Design

This research scenario uses a mixed-method approach combining qualitative and quantitative methods:

- 1. Qualitative Approach:**

- a. The review of the available literature and other case studies will serve as bases for problem identification in supply chains.

- b. The model will be a unique framework for the input of ZTA into supply chains.

2. Quantitative Approach:

- a. Integration of ZTA with tools like Kubernetes, Vault, Spire, and Istio in a simulated supply chain environment.
- b. Performance measurement using metrics such as latency, throughput, and resource utilization.

3.2 Data Collection

Data is collected from different sources for comprehensive analysis:

1. Primary Data:

- a. Performance metrics obtained from the implemented ZTA setup.
- b. Prometheus and Grafana used for logs and monitoring data.
- c. SPIRE and Istio for access control and policy violations metrics

2. Secondary Data:

- a. Literature on the Zero Trust principles, supply chain vulnerabilities and the existing security architectures.
- b. Case studies that examine the companies that have adopted Zero Trust or other related architectures.

3.3 Implementation

This phase follows an iterative process to deploy ZTA in a supply chain environment, as described below:

1. Infrastructure Setup:

- a. Deploying a Kubernetes clusters to represent supply chain domains (e.g., Procurement, Logistics, Inventory, Distribution).
- b. Integrating HashiCorp Vault, SPIRE, and Istio for security, identity management, and traffic control.

2. Policy Design and Enforcement:

- a. Open Policy Agent (OPA) and Istio Authorization Policies to define Zero Trust policies for access control and resource usage.

3. Monitoring and Observability:

- a. Deploying Prometheus and Grafana for monitoring system performance and visualizing metrics.

4. Testing and Validation:

- a. Simulating legitimate and malicious traffic to test the effectiveness of ZTA.
- b. Measuring key performance indicators (KPIs) such as latency, access control efficiency, and resource utilization.

3.4 Tools and Technologies

The following tools and technologies were used for implementation and evaluation:

- **Infrastructure:** Kubernetes, Minikube
- **Security and Identity:** HashiCorp Vault, SPIRE
- **Policy Enforcement:** Open Policy Agent (OPA), Istio
- **Monitoring and Observability:** Prometheus, Grafana
- **Networking:** Istio Service Mesh
- **Development and Deployment:** Helm, Docker

3.5 Data Analysis

The collected data is analysed to evaluate the effectiveness of ZTA in mitigating risks and improving security in supply chains:

1. Qualitative Analysis:

- a. Evaluate the adaptability of ZTA principles to supply chain scenarios.
- b. Compare ZTA with traditional security approaches in terms of coverage and resilience.

2. Quantitative Analysis:

- a. Metrics such as response time, resource consumption, and number of blocked unauthorized access attempts are used to measure the performance.
- b. Visualizations generated via Grafana dashboards provide insights into system behaviour.

3.6 Validation

The implemented ZTA is validated against key objectives:

- **Security:** Ensuring any unauthorized access is blocked at all levels (e.g., service-to-service communication, user access).
- **Performance:** Evaluating the trade-offs introduced by ZTA, such as latency and resource overhead.
- **Scalability:** Test the framework's ability to handle increasing workloads and additional supply chain components.

3.7 Ethical Considerations

The following ethical concerns are addressed by:

- Following standard best practices to handle any proprietary tools or methods.
- Ensuring the simulated environment is secure and does not expose sensitive real-world supply chain data.

3.8 Limitations

Even though this methodology provides a framework for evaluating ZTA in supply chains, there are a few limitations:

- **Simulation Environment:** Real-world supply chain complexities may not be fully replicated.
- **Resource Constraints:** Limited resources may impact scalability testing.

4 Design Specification

This phase gives a detailed design specification of the architectural framework, tools, and processes used to implement ZTA in a supply chain which includes core principles, system components, and their interactions. The design follows the principle of “never trust, always verify”. The design is structured to enforce identity-based authentication, least privilege access, and real-time monitoring across all components of the supply chain, including procurement, logistics, inventory, and distribution.

4.1 Architecture Overview

The framework is implemented using a microservices-based architecture deployed in Kubernetes. The system includes the following components:

4.1.1 Core Components

- 1. Identity Management:**
 - a. SPIRE is used for issuing SPIFFE IDs to workloads for secure identity verification.
 - b. HashiCorp Vault manages secrets and credentials for service-to-service and user authentication.
- 2. Policy Enforcement:**
 - a. Open Policy Agent (OPA) defines and enforces access policies.
 - b. Istio AuthorizationPolicies restrict communication between microservices based on roles and identities.
- 3. Networking and Traffic Control:**
 - a. Istio service mesh secures traffic between services, provides mutual TLS (mTLS) encryption and observability.
- 4. Monitoring and Logging:**
 - a. Prometheus and Grafana collects and visualizes the performance metric.
 - b. Logs are collected using tools like Fluentd or Loki for centralized analysis.

4.1.2 Supply Chain Microservices

Each component of the supply chain represents a Kubernetes microservice:

- 1. Procurement Service:**
 - a. Handles purchase orders and vendor interactions.
 - b. Access restricted to authorized users and services like inventory.
- 2. Logistics Service:**
 - a. Manages shipment tracking and delivery schedules.
 - b. Communicates securely with distribution and inventory.
- 3. Inventory Service:**
 - a. Tracks stock levels and triggers restocking processes.

- b. Shares data with procurement and logistics.
- 4. **Distribution Service:**
 - a. Handles order fulfilment and customer deliveries.
 - b. Connects with logistics to optimize routing.

4.2 Security

1. **Secure Communication:**
 - a. Istio mTLS provides encrypted communication between services.
 - b. The security rules are managed by Istio Gateway which also manages the ingress and egress traffic.
2. **Authentication and Identity:**
 - a. SPIRE is used to issue cryptographically verifiable SPIFFE IDs for workloads.
 - b. Vault provides secrets and dynamic credentials for database access.
3. **Access Control:**

Istio AuthorizationPolicies and OPA enforces polices:

 - a. User-Level: Enforces RBAC.
 - b. Service-Level: Based on polices it restricts communication between services.

4.3 Deployment

4.3.1 Kubernetes Cluster

Using MiniKube, the architecture is deployed in Kubernetes using different namespaces which separate the supply chain components (procurement, logistics, inventory, distribution).

4.3.2 Tools

1. **HashiCorp Vault:**
 - a. Stores credentials like database passwords and API keys.
 - b. Injects secrets into Kubernetes pods via annotations.
2. **Istio Service Mesh [17]:**
 - a. Secures inter-service communication.
 - b. Provides telemetry data for monitoring.
3. **Prometheus and Grafana:**
 - a. Monitor resource usage and performance metrics.
 - b. Provide dashboards for real-time observability.
4. **Open Policy Agent (OPA):**
 - a. Enforces policies such as restricting logistics service access to specific namespaces.

4.4 Data Flow Design

4.4.1 Request Flow

1. User or service sends a request to a microservice (e.g., procurement).
2. The Istio Gateway validates the request and forwards it to the appropriate service.
3. SPIRE authenticates the service identity via SPIFFE ID.
4. OPA checks access policies before the request is processed.
5. If authorized, the request is handled by the target service.

4.4.2 Secrets Flow

1. Vault injects secrets (e.g., database credentials) into pods.
2. The application retrieves secrets from the injected file or environment variables.

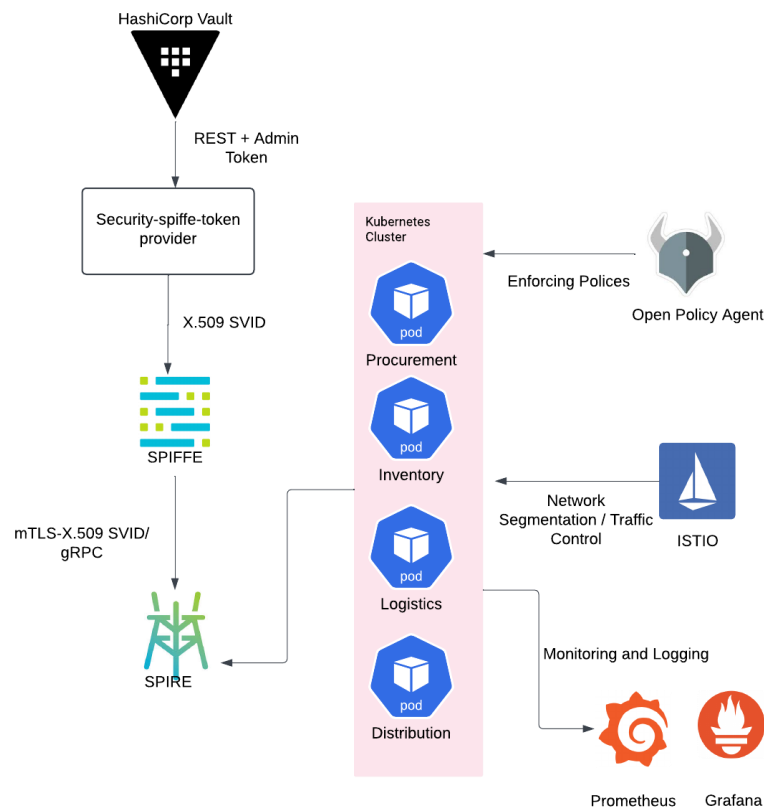


Fig 1. Proposed Architecture Diagram.

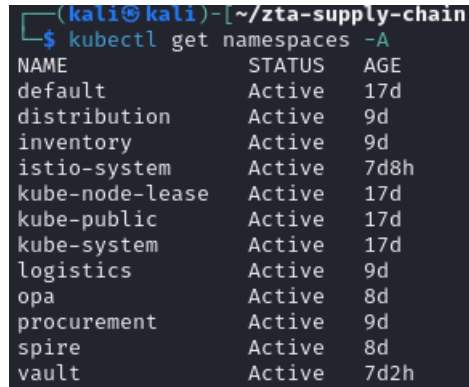
5 Implementation

The implementation of ZTA includes setting up a secure, identity-driven, and policy-enforced environment using Kubernetes, Istio, SPIRE, HashiCorp Vault, Open Policy Agent (OPA), and Prometheus-Grafana for monitoring.

5.1 Infrastructure Setup

- Using Minikube, a Kubernetes cluster was created to simulate an environment of supply chain

- Separate namespaces were created to showcase different supply chain components (e.g. procurement, logistics, inventory, distribution).



```
(kali@kali)~[zta-supply-chain]
$ kubectl get namespaces -A
NAME                STATUS   AGE
default             Active   17d
distribution         Active   9d
inventory            Active   9d
istio-system        Active   7d8h
kube-node-lease      Active   17d
kube-public          Active   17d
kube-system          Active   17d
logistics            Active   9d
opa                  Active   8d
procurement          Active   9d
spire                Active   8d
vault                Active   7d2h
```

Fig 2. Namespaces for different services

- Each of the supply chain component was deployed as a microservice in its respective namespace using a deployment.yaml and service.yaml file.

5.2 Identity Management Using SPIRE

- SPIRE was deployed for the server as a StatefulSet and for the agent as a DaemonSet [15].
- The configuration includes defining trusted domains and registration entries for services.



```
$ cat procurement.json
{
  "spiffeId": "spiffe://example.org/procurement",
  "selector": {
    "type": "k8s",
    "value": "ns:procurement"
  }
}
```

Fig 3. Registering entries for SPIRE.

5.3 Policy Using OPA

- An admission controller for Kubernetes is used as Open Policy Agent (OPA).
- Policies are written in Rego to implement access control.
- The policies are loaded into OPA and validated via Kubernetes Admission Webhooks [16].

```

$ cat opa-config.yaml
apiVersion: v1
kind: ConfigMap
metadata:
  name: opa-policies
  namespace: opa
data:
  allow-procurement.rego: |
    package kubernetes.admission

    default allow = false

    allow {
      input.request.namespace = "procurement"
      input.request.operation = "CREATE"
      input.request.kind.kind = "Pod"
    }

  allow-logistics.rego: |
    package kubernetes.admission

    default allow = false

    allow {
      input.request.namespace = "logistics"
      input.request.operation = "CREATE"
      input.request.kind.kind = "Pod"
    }

```

Fig 4. OPA Configuration for different components of supply chain.

5.4 Service Mesh Using Istio

- Istio is installed using the “istioctl” with the demo profile to unlock all core features.
- mTLS is configured for secure communications between the services.
- A Istio Gateway is deployed to handle ingress and egress traffic.

```

$ cat allow-procurement.yaml
apiVersion: security.istio.io/v1beta1
kind: AuthorizationPolicy
metadata:
  name: allow-procurement
  namespace: procurement
spec:
  rules:
  - from:
    - source:
      namespaces: ["logistics", "inventory"]

```

Fig 5. Authorization policies for different components of supply chain.

5.5 Secret Management Using HashiCorp Vault

- The vault is deployed as a StatefulSet in Kubernetes.
- Secrets are managed using the Vault KV secrets engine.
- Secrets are injected into the pods via Vault annotations in the deployment.yaml file of the supply chain component.
- Vault policies are defined to restrict access to specific secrets.

```

$ cat procurement-policy.hcl
path "secret/data/procurement/*" {
  capabilities = ["read", "list"]
}

```

Fig 6. Policies to allow for secrets injection.

5.6 Monitoring and Observability

- Prometheus is deployed for metrics collection and Grafana for visualization.
- Metrics endpoints are deployed by microservices at “/metrics”.
- Pre-configured Kubernetes and Istio dashboards are imported into Grafana for real-time monitoring.
- ServiceMonitors were defined for Prometheus to scrape metrics from specific services.

5.7 Testing and Validation

1. **Security Tests:**
 - a. Simulated unauthorized access attempts between namespaces to validate that Istio AuthorizationPolicies and OPA policies are deployed correctly.
 - b. Verified mTLS encryption for all inter-service communication.
2. **Performance Tests:**
 - a. Measured response times, CPU usage, and memory utilization using Prometheus.
3. **Secrets Management:**
 - a. Confirmed that secrets were securely injected into pods and inaccessible to unauthorized services.

6 Evaluation

This section provides the outcomes of the proposed ZTA implementation in the supply chain which aligns with the objectives of the research.

6.1 Test Cases

6.1.1 Security

1. **Access Control:**
 - a. Attempting to access logistics services via unapproved namespace.
 - b. Verifying the Istio AuthorizationPolicies blocks the request.
2. **Secrets Management:**
 - a. Testing and verifying unauthorized access to Vault secrets is denied.
 - b. Ensuring proper injection into the authorised pods
3. **Identity Verification:**
 - a. Ensure SPIRE denies invalid workload identities.

6.1.2 Performance

1. Monitor resource utilization (CPU, memory) under different workloads of Istio sidecars, Vault, and SPIRE.
2. Evaluating time taken for policy enforcement and secrets injection.
3. Measure average request latency before and after ZTA implementation.

6.1.3 Scalability

1. Integrate new namespaces and services into the process and confirm the system's capability in adding them.
2. Run high traffic to guarantee that the ZTA framework caters to numerous requests without much degradation in performance.

6.2 Comparison

The proposed implementation was deployed locally compared to the ones in the managed Kubernetes Service like AWS Elastic Kubernetes Services (EKS) and Azure Kubernetes Services (AKS). The following comparison highlights the differences:

Feature	Proposed Implementation	AWS EKS	AZURE AKS
Cost	Self-managed.	High cost due to AWS managed services and using additional tools.	Managed pricing model, but costs increase with Azure-specific integrations.
Scalability	Change in manual configuration helps in Kubernetes cluster scalability.	AWS feature like auto-scaling helps in this.	Azure Auto-scaler helps in scaling but requires cloud dependencies.
Zero Trust Support	Fully customisable.	It is limited due to additional integrations (e.g., AWS IAM, App Mesh).	Limited as it relies on external tools and Azure AD.
Identity Management	SPIFFE IDs for workload identity issued by SPIRE	AWS IAM and EKS service accounts (not SPIFFE-compatible).	Pod identity through Azure Workload Identity and Azure AD for user-level identity.
Secrets Management	HashiCorp Vault manages the secrets securely and dynamically.	Using AWS Secrets Manager which requires manual or semi-automated setup.	Azure Key Vault can be utilised with additional configuration.
Service Mesh	Istio provides traffic control, mTLS, and observability.	AWS App Mesh supports mTLS but lacks policies like Istio.	Open Service Mesh is a feature that's available and less feature rich.

Monitoring and Observability	Prometheus and Grafana provide detailed metrics and visualizations.	CloudWatch and AWS X-Ray are native but require more configuration for deep insights.	Azure Monitor and Log Analytics provide native monitoring but are less customizable.
Policy Enforcement	OPA enforces policies for all the service defined	Using Kyverno or AWS WAF	AZURE policies are limited to specific configurations and resources.
Cloud Dependence	Portable across different environments and it is cloud-agnostic	Fully dependent on AWS Cloud	Fully dependent on AZURE Cloud

Table 1. Comparison of ZTA in different services

6.3 Discussion

1. Full Zero Trust Compliance:

By using SPIRE, a deviation from tradition would occur and it would be introduced to the solution in the form of cryptographically secure identities (SPIFFE IDs) for workloads, which are portable and cloud-agnostic.

- Workload Identity:** With SPIRE, each of the workloads is represented as a fixed set of verifiable information in the form of a cryptographically secure identity (SPIFFE IDs) which is portable and cloud-agnostic.
- Access Policies:** The combination between Open Policy Agent (OPA), an already popular and well-tested policy engine and a highly granular approach to policies is the key reason for which the supply chain microservices are perfectly compatible and can be seamlessly integrated with security tools for supply chain microservices.
- Secrets Management:** HashiCorp Vault dynamically handles secrets, making it more robust than static solutions like AWS Secrets Manager or Azure Key Vault.

2. Greater Flexibility:

When compared to AWS EKS and Azure AKS, the current implementation can be seen as a chance to enhance the system without being tied to a certain provider:

- It can be run on-premises, in a hybrid cloud, or even in a multi-cloud setup. Thus, it is suitable even for those organizations that highly regard the vendor neutrality aspect.
- For a supply chain to operate and interact with components situated in multiple regions or set up in different infrastructure, this flexibility is very important.

3. Superior Monitoring and Observability:

Prometheus and Grafana give much better choices of customization for monitoring compared to the cloud-based solutions as follows:

- a. Supply chains get reviewed and ensure the expected security/performance needs through specific designs of monitoring and dashboards.
- b. Incident detection and response become much quicker by means of tools like log and alerts that are highly customizable.

4. Cost-Efficiency:

- a. Avoiding managed services such as AWS EKS and Azure AKS can cut down a high-cost bill in a self-managed approach.
- b. Tools like SPIRE, Istio, and Vault have been the means to zeroing the costs on licenses and the maintenance of the function in the long run.

5. Cloud-Agnostic Scalability:

- a. While AWS and Azure provide robust auto-scaling, their dependence on cloud-specific configurations limits portability:
- b. The custom implementation leverages Kubernetes-native scaling and can be adapted to any cloud provider or on-premises hardware.

6. Enhanced Service Mesh Capabilities:

Istio offers more features compared to AWS App Mesh or Azure's Open Service Mesh:

- a. Optimal connection with Prometheus and Grafana.
- b. Advanced traffic routing and mTLS security check for all microservices.

7 Conclusion and Future Work

7.1 Conclusion

The implementation of ZTA in a simulated supply chain environment was a challenging approach to demonstrate the effectiveness of applying advanced security principles in securing critical infrastructure. The complete implementation was not successful but still the study provided important insights into the challenges and opportunities with adopting ZTA in supply chain management:

1. Complex Integration:

- a. To integrate the various tools such as Istio, Vault, and SPIRE into a Kubernetes environment becomes challenging due to the steep learning curve, complex configurations, and tool-specific dependencies.
- b. Balancing the interoperability of these tools with the performance of the system was another big issue.

2. Infrastructure Limitations:

- a. The absence of a true production environment in the real world made it impossible to fully test the architecture's scalability, reliability, and resilience of the system.
- b. Persistent volume claims, pod scheduling issues, and monitoring setups required a great deal of configuration work.

3. Significant Benefits of ZTA Principles:

- a. Although it was a part of the implementation stage, the principles of ZTA were properly demonstrated when features like identity-based authentication, least privilege access, and secure communication were enabled in isolated components.
 - b. Security tools such as SPIRE and Istio were very promising in establishing a security framework when rightfully configured.
4. **Challenges in Monitoring and Observability:**
- a. The configuration mismatches and the resource constraints considerably challenged the setting up of meaningful metrics collection and visualization in Prometheus and Grafana.

7.2 Future works

To address the challenges faced in this research, the following future works is recommended:

1. **Simplified Deployment Model:**
 - a. Develop a process through using Helm charts or deployment scripts to reduce the complexity of setting and integrating multiple tools.
2. **Testing Environment:**
 - a. Create a test environment which closely simulates a real-world supply chain, including varied traffic patterns, larger workloads.
 - b. Integrating CI/CD pipelines to validate the ZTA framework via automation testing.
3. **Tool Specific Enhancement:**
 - a. Research advanced Istio configurations for greater performance and policy enforcement.
 - b. Investigate alternative or improvements for workload identity in SPIRE.
4. **Adopting Managed Services:**
 - a. Making use of other Kubernetes offering like AWS EKS and AZURE AKS to reduce operational overhead whilst integrating custom ZTA tools for security.
 - b. Test with cloud-native tools like AWS App Mesh or Azure Workload Identity for better integration.
5. **Better Monitoring and Observability:**
 - a. Automating the dashboards and alerts in Grafana for better insights into system performance and security metrics.
 - b. Inclusion of tools like Loki for centralization of log aggregation.
6. **Policy Optimisation:**
 - a. Configure Open Policy Agent (OPA) for security and maximize speed.
 - b. Audit policies on automation to identify incorrect configurations and ascertain compliance with Zero Trust policies.

References

1. Amaral, T., & Gondim, J., 2021. Integrating Zero Trust in the cyber supply chain security. *2021 Workshop on Communication Networks and Power Systems (WCNPS)*, pp. 1- 6. <https://doi.org/10.1109/WCNPS53648.2021.9626299>.
2. Collier, Z., & Sarkis, J., 2021. The zero trust supply chain: Managing supply chain risk in the absence of trust. *International Journal of Production Research*, 59, pp. 3430 - 3445. <https://doi.org/10.1080/00207543.2021.1884311>.
3. Pan, Y., Liu, S., Zhang, J., & Wu, H., 2023. A Supply Chain Finance Framework Based on Zero-Trust Architecture. *2023 IEEE 14th International Conference on Software Engineering and Service Science (ICSESS)*, pp. 217-221. <https://doi.org/10.1109/ICSESS58500.2023.10293068>.
4. Rose, S., Borchert, O., Mitchell, S. and Connelly, S. (2020), Zero Trust Architecture, Special Publication (NIST SP), *National Institute of Standards and Technology*, Gaithersburg, MD, [online], <https://doi.org/10.6028/NIST.SP.800-207>
5. Teerakanok, S., Tang, M., & Kazerani, M. (2021). Migrating to Zero Trust Architecture: Reviews and Challenges. *Security and Communication Networks*. <https://onlinelibrary.wiley.com/doi/10.1155/2021/8811138>
6. Khan, M., 2023. Zero trust architecture: Redefining network security paradigms in the digital age. *World Journal of Advanced Research and Reviews*. <https://doi.org/10.30574/wjarr.2023.19.3.1785>.
7. Hosney, E., Halim, I., & Yousef, A., 2022. An Artificial Intelligence Approach for Deploying Zero Trust Architecture (ZTA). *2022 5th International Conference on Computing and Informatics (ICCI)*, pp. 343-350. <https://doi.org/10.1109/icci54321.2022.9756117>.
8. Edo, O., Tenebe, T., Etu, E., Ayuwu, A., Emakhu, J., & Adebisi, S., 2022. Zero Trust Architecture: Trend and Impact on Information Security. *International Journal of Emerging Technology and Advanced Engineering*. https://doi.org/10.46338/ijetae0722_15.
9. Blancaflor, E., Abat, A., Degrano, K., Lindio, M., & Pamoso, A., 2023. Implementation of Zero Trust Security to Reduce Ransomware Attacks in the Philippines: A Literature Review. *Journal of Advances in Information Technology*. <https://doi.org/10.12720/jait.14.5.928-933>.
10. Phiayura, P., & Teerakanok, S., 2023. A Comprehensive Framework for Migrating to Zero Trust Architecture. *IEEE Access*, 11, pp. 19487-19511. <https://doi.org/10.1109/ACCESS.2023.3248622>.
11. N. F. Syed, S. W. Shah, A. Shaghaghi, A. Anwar, Z. Baig and R. Doss, "Zero Trust Architecture (ZTA): A Comprehensive Survey," in *IEEE Access*, vol. 10, pp. 57143-57179, 2022, doi: 10.1109/ACCESS.2022.3174679

12. Paul, B. and Rao, M. (2022) 'Zero-Trust Model for Smart Manufacturing Industry', *Applied Sciences*, 13, p. 221. Available at: <https://doi.org/10.3390/app13010221>.
13. Gupta, V., & Bharathi, S. V. (2023). Supply Chain Risk Management Through Zero-Trust Architecture. *Springer International Conference Proceedings*.
14. Van Bossuyt, D. L., Hale, B., & Arlitt, R. (2023). Zero-trust for the system design lifecycle. *Journal of Computing and Engineering*
15. Spiffe.io. (2024). *SPIFFE / Quickstart for Kubernetes*. [online] Available at: <https://spiffe.io/docs/latest/try/getting-started-k8s/>
16. Guest Expert (2022). *Open Policy Agent with Kubernetes - Tutorial (Pt. 1)*. [online] GitGuardian Blog - Take Control of Your Secrets Security. Available at: <https://blog.gitguardian.com/open-policy-agent-with-kubernetes-tutorial-pt-1/>.
17. Tetrate. (2021). *What Is Istio and Why Does Kubernetes Need it?* [online] Available at: <https://tetrate.io/blog/what-is-istio-and-why-does-kubernetes-need-it/>.