Aalto University
School of Science

Master's Programme in Security and Cloud Computing

# Securing Amazon Web Services with Zero Trust Architecture

**Shweta Jaiswal**

Master's Thesis
2024

**Aalto University**
**School of Science**

**Author** Shweta Jaiswal

**Title** Securing Amazon Web Services with Zero Trust Architecture

**Degree programme** Master's Programme in Security and Cloud Computing

**Major** Security and Cloud Computing

**Supervisor** Prof. Yuming Jiang, Prof. Mario Di Francesco

**Advisor** Thomas Lund

**Collaborative partner** Sikt, Norway

**Date** 13 July 2024 **Number of pages** 92 **Language** English

**Abstract**

Securing cloud infrastructure has become a crucial priority for organizations worldwide in the rapidly evolving technological landscape. As businesses increasingly depend on cloud-based systems, safeguarding sensitive information against potential threats is essential. Amazon Web Services (AWS) is one of the public cloud providers offering various services for organizations. Despite its widespread adoption, AWS faces significant security challenges, evidenced by notable breaches such as the Capital One and Imperva incidents. These breaches underscore the need for proactive security measures to address the security challenges in AWS.

Traditional security models, such as perimeter-based or trust-based, are insufficient for protecting AWS environments due to the dynamic nature of cloud resources and the porous network perimeters. Zero Trust Architecture (ZTA) provides a more robust approach by operating under the assumption that threats can originate inside and outside the network. It advocates for granular access controls, micro-segmentation, and continuous authentication and authorization to minimize the attack surface and prevent lateral movement within the network.

This thesis focuses on enhancing the security of AWS using ZTA. It comprehensively reviews the state-of-the-art techniques for implementing ZTA tenets and addresses AWS security challenges. It offers insights and solutions to enhance AWS security through the adoption of ZTA principles. Furthermore, it presents a Proof-of-Concept (POC) implementation of a ZTA system to securely manage AWS resources. The POC implementation was conducted in collaboration with Sikt - The Norwegian Infrastructure Provider for Research and Education.

**Keywords** Zero Trust Architecture, Amazon Web Services Security, cloud security, ZTA Implementation , ZTA Challenges

**Title:** Securing Amazon Web Services with Zero Trust Architecture
**Student:** Jaiswal, Shweta

**Problem description:**

In recent years, cloud computing has become an integral component of modern Information Technology (IT) infrastructure, enabling businesses to scale operations and leverage advanced technologies. Amazon Web Services (AWS) is one of the most widely used cloud platforms in the world, providing comprehensive services to help organizations lower costs, increase agility, and innovate faster. However, as cloud adoption grows, so does the need for robust security measures to protect sensitive data and applications hosted in the cloud.

Zero Trust Architecture (ZTA), characterized by continuous verification and authentication principles, offers a paradigm shift from traditional perimeter-based security models. ZTA aims to reduce a network's attack surface by segmenting it into smaller, more manageable areas, and by implementing strict access controls that limit the movement of threats within the network. By doing so, ZTA lowers the risk of a data breach and helps organizations to detect and respond to security incidents more quickly.

The purpose of this project is to explore enhancing cloud security by implementing ZTA. The focused cloud platform is AWS. Specifically, we first comprehensively examine security vulnerabilities that arise within AWS environments while deploying applications and systems. Then, we discuss the possible solutions to these problems and propose implementing zero-trust principles to mitigate the issues. Additionally, if the time permits we will implement zero trust network to address challenges such as secure management of Amazon Relational Database Service (RDS) and Virtual Machines (VMs) (This will be implemented in collaboration with the Norwegian ICT Infrastructure Provider - Sikt). Furthermore, we evaluate the zero-trust solutions and discuss the limitations to provide a holistic understanding of their effectiveness in enhancing security measures within AWS environments.

**Approved on:** 2024-02-05
**Main supervisor:** Jiang, Yuming, NTNU
**Co-supervisors:** Di Francesco, Mario, Aalto University and Lund, Thomas, Sikt

# Abstract

Securing cloud infrastructure has become a crucial priority for organizations worldwide in the rapidly evolving technological landscape. As businesses increasingly depend on cloud-based systems, safeguarding sensitive information against potential threats is essential. AWS is one of the public cloud providers offering various services for organizations. Despite its widespread adoption, AWS faces significant security challenges, evidenced by notable breaches such as the Capital One and Imperva incidents. These breaches underscore the need for proactive security measures to address the security challenges in AWS.

Traditional security models, such as perimeter-based or trust-based, are insufficient for protecting AWS environments due to the dynamic nature of cloud resources and the porous network perimeters. ZTA provides a more robust approach by operating under the assumption that threats can originate inside and outside the network. It advocates for granular access controls, micro-segmentation, and continuous authentication and authorization to minimize the attack surface and prevent lateral movement within the network.

This thesis focuses on enhancing the security of AWS using ZTA. It comprehensively reviews the state-of-the-art techniques for implementing ZTA tenets and addresses AWS security challenges. It offers insights and solutions to enhance AWS security through the adoption of ZTA principles. Furthermore, it presents a Proof-of-Concept (POC) implementation of a ZTA system to securely manage AWS resources. The POC implementation was conducted in collaboration with Sikt - The Norwegian Infrastructure Provider for Research and Education.

## Preface

I want to express my heartfelt gratitude to my family for standing by my side and encouraging me through the difficult times. Their unwavering support has been invaluable throughout this journey. I am also deeply thankful to my team at Sikt for their immense support and understanding during the whole process. Last but not least, I extend my sincere thanks to my supervisors. This thesis would not have been possible without their guidance and mentorship.

With the support of the
Erasmus+ Programme
of the European Union

# Contents

# List of Figures

# List of Tables

# List of Acronyms

**2FA** Two-Factor Authentication.

**ABAC** Attribute-Based Access Control.

**ABE** Attribute-Based Encryption.

**ACE** Access Control Engine.

**ACLs** Access Control Lists.

**AI** Artificial Intelligence.

**API** Application Programming Interface.

**APT** Advanced Persistent Threat.

**AU-ZTAC** Attribute and User Trust Score-Based Zero Trust Access Control Model.

**AWS** Amazon Web Services.

**BYOD** Bring-Your-Own-Device.

**CA** Certificate Authority.

**CASA** Context-Aware Scalable Authentication.

**CGNAT** Carrier-Grade Network Address Translation.

**CISO** Chief Information Security Officer.

**CP-ABE** Ciphertext-Policy Attribute-Based Encryption.

**CSA-SDP** Cloud Security Alliance's Software Defined Perimeter.

**D2D** Device-to-Device.

**DDoS** Distributed Denial-of-Service.

**DNS** Domain Name System.

**DNSSEC** Domain Name System Security Extensions.

**DoS** Denial-of-Service.

**dRBAC** Distributed Role-Based Access Control.

**eBPF** extended Berkeley Packet Filter.

**EBS** Elastic Block Store.

**EC2** Elastic Compute Cloud.

**EU** European Union.

**FBAC** Function-Based Access Control.

**FURZE** Fuzzy Risk Framework for Zero Trust Network.

**GCP** Google Cloud Platform.

**HE** Homomorphic Encryption.

**HTTP** HyperText Transfer Protocol.

**HTTPS** HyperText Transfer Protocol Secure.

**IaaS** Infrastructure-as-a-Service.

**IaC** Infrastructure-as-Code.

**IAM** Identity and Access Management.

**IBAC** Identity-Based Access Control.

**IBE** Identity-Based Encryption.

**IDS** Intrusion Detection Systems.

**IIoT** Industrial Internet of Things.

**IoT** Internet of Things.

**IoV** Internet of Vehicles.

**IP** Internet Protocol.

**IPFS** Interplanetary File System.

**IT** Information Technology.

**KMS** Key Management Service.

**KP-ABE** Key-Policy Attribute-Based Encryption.

**L-ABAC** Location-Aware Attribute-Based Access Control.

**LCDA** Lightweight Continuous Device-to-Device Authentication.

**LDAP** Lightweight Directory Access Protocol.

**LFAs** Link Flooding Attacks.

**MFA** Multi-Factor Authentication.

**MitM** Man-in-the-Middle.

**NAT** Network Address Translation.

**NFTs** Non-Fungible Tokens.

**NFV** Network Function Virtualization.

**NIST** National Institute of Standards and Technology.

**NTNU** Norwegian University of Science and Technology.

**OS** Operating System.

**OTP** One-Time Password.

**PA** Policy Administrator.

**PaaS** Platform-as-a-Service.

**PAM** Privileged Access Management.

**PDP** Policy Decision Point.

**PE** Policy Engine.

**PEP** Policy Enforcement Point.

**PKG** Private Key Generator.

**PKI** Public Key Infrastructure.

**POC** Proof-of-Concept.

**RBAC** Role-Based Access Control.

**RBE** Role-Based Encryption.

**RDS** Relational Database Service.

**RiBAC** Risk-Based Access Control.

**RRABE** Re-Randomizable Attribute-Based Encryption.

**RS-HABE** Revocable-Storage Hierarchical Attribute-Based Encryption.

**RS-IBE** Revocable-Storage Identity-Based Encryption.

**S3** Simple Storage Service.

**SaaS** Software-as-a-Service.

**SADAC** Security Attribute-Based Dynamic Access Control.

**SASE** Secure Access Service Edge.

**SDG** Sustainable Development Goal.

**SDGs** Sustainable Development Goals.

**SDN** Software-Defined Networking.

**SDP** Software-Defined Perimeter.

**SIEM** Security Information and Event Management.

**SPA** Single Packet Authentication.

**SQL** Structured Query Language.

**SSE** Server-Side Encryption.

**SSH** Secure Shell.

**SSL** Secure Sockets Layer.

**SSO** Single Sign-On.

**TDE** Transparent Data Encryption.

**TLS** Transport Layer Security.

**TPM** Trusted Platform Module.

**T-RBAC** Task-Role-Based Access Control.

**UAP** Unified Authentication Platform.

**UN** United Nations'.

**VEN** Virtual Enforcement Node.

**VLAN** Virtual Local Area Network.

**VM** Virtual Machine.

**VMMs** Virtual Machine Monitors.

**VMs** Virtual Machines.

**VPC** Virtual Private Cloud.

**VPN** Virtual Private Network.

**WAF** Web Application Firewall.

**XSS** Cross-Site Scripting.

**ZAIB** Zero Trust and Attribute-Based Access Control for Internet of Things using Blockchain.

**ZT** Zero Trust.

**ZTA** Zero Trust Architecture.

**ZTN** Zero Trust Network.

**ZTNA** Zero Trust Network Access.

**ZTX** Zero Trust eXtended.

# Chapter 1
# Introduction

In today's rapidly evolving world of technology, securing cloud infrastructure has become a crucial priority for organizations worldwide. As businesses increasingly depend on cloud-based systems, protecting sensitive information against potential threats is essential. Major public cloud providers such as AWS, Google Cloud Platform (GCP), and Microsoft Azure play a major role in meeting the diverse needs of organizations. The choice of migrating to the cloud depends on cost, adoption, security, and trust in the cloud provider to keep the data safe.

AWS has increasingly become a preferred choice for companies seeking to leverage the benefits of the cloud. It is one of the prominent cloud providers, with a market share of 31% in Q4 2023 [1] and provides three major categories of services: Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS), and Software-as-a-Service (SaaS) [2]. It offers flexible computing resources that allow businesses to easily adapt their infrastructure to meet the customers' changing needs. AWS operates on a pay-as-you-go pricing model, so organizations are only charged for the resources they use. It helps businesses save on capital expenditures, making it an appealing choice for new startups and established enterprises. Some companies that rely heavily on AWS for their services include Netflix, LinkedIn, and Facebook.

The adoption of AWS has seen remarkable growth and is expected to continue. However, with the growth comes the need to address security challenges. Safeguarding AWS infrastructure and services involves understanding the necessary methods and practices to protect data, applications, and networks. Neglecting to implement effective security measures can lead to serious security breaches, as seen in some notable cases [3]. In July 2019, Capital One experienced a data breach affecting 100 million customers. It was attributed to a former Amazon employee exploiting a misconfigured open-source Web Application Firewall (WAF) on Capital One's AWS servers [4]. In October 2019, Imperva revealed a data breach originating from a misconfigured AWS instance, granting attackers access to a RDS snapshot containing customer credentials, highlighting the risk posed by application and infrastructure misconfigurations [5].

The security breaches highlight the urgent need for organizations to proactively address security concerns and vulnerabilities that emerge from their utilization of AWS services. The security challenges can be categorized into different aspects: IAM, data security, network security, virtualization security, web application and API security, multi-tenancy, and visibility. Previously, traditional perimeter-based and trust-based security have been used for safeguarding AWS environments. However, their efficacy is limited in the current security landscape. The dynamic nature of resource allocation, porous network perimeters, and subjective trust assessments in cloud environments introduce risks of misconfigurations and oversight of critical security factors, such as authentication vulnerabilities and insider threats, which traditional methods cannot adequately address. As a result, a more sophisticated security paradigm is imperative.

ZTA emerges as a robust response for organizations moving away from traditional perimeter defenses and focusing on internal security. It operates under the premise that threats may arise from internal and external sources, advocating for continuous verification and stringent access controls irrespective of user location or network segment [6]. ZTA diverges from the reliance on perimeter defenses, opting instead for granular access controls that minimize the attack surface and thwart lateral movement by potential intruders within the network. Micro-segmentation is a key tenet of ZTA, where the network is subdivided into smaller, more manageable segments, and access controls are enforced at the application or workload level. This approach tailors policies according to user identity, application context, and dynamic variables, ensuring finer-grained access control [7].

Moreover, ZTA facilitates seamless management and enforcement of access policies within dynamic cloud environments [8]. By implementing continuous authentication and authorization, ZTA dynamically evaluates access decisions based on user identity, device health, location, and behavior [9]. This dynamic assessment mitigates risks associated with credential theft, privilege escalation, and insider threats [10] by adhering to the principle of least privilege and maintaining ongoing verification. In essence, ZTA fortifies security measures and streamlines access management in contemporary network infrastructures. As more and more businesses migrate to AWS, it is equally vital to apply ZTA principles to safeguard their AWS resources.

This thesis focuses on improving the security of AWS through ZTA. The thesis aims to thoroughly review the current state-of-the-art techniques for implementing ZTA. Furthermore, it discusses the security challenges of AWS environments and proposes how these challenges can be addressed through ZTA. Additionally, it presents a POC implementation of a zero trust system for securely accessing and managing

AWS resources at Sikt[1]. The work endeavors to address the following research questions:

- Q1: What is the current state-of-the-art research to implement the tenets of ZTA?
- Q2: What are the security challenges present in AWS?
- Q3: How can the principles of ZTA be used to enhance the security of AWS environments?

Our main contributions include:

- A comprehensive review of the current state-of-the-art techniques for realizing the tenets of ZTA. The techniques include authentication, access control, encryption, network segmentation, automation and orchestration, and monitoring and analytics. Furthermore, it includes a discussion on the challenges in the implementation of ZTA.
- An in-depth exploration of the security challenges present in AWS environments and discussion on how the principles of ZTA can be leveraged to enhance the security of AWS environments. Furthermore, it includes an analysis of the various solutions prevalent in the industry to realize ZTA in AWS environments.
- A POC implementation of a zero trust system to securely access and manage AWS resources such as RDS and Elastic Compute Cloud (EC2) instances. Additionally, there is a discussion on how this POC can be adopted by organizations such as Sikt.

This thesis is structured as follows: Chapter 2 provides the background on different types of clouds, AWS services, the basic principles of ZTA, and relevant real-world implementations. Chapter 3 presents the state-of-the-art research for the realization of ZTA principles. It also provides a discussion on various challenges in the implementation of ZTA along with a qualitative evaluation of the ZTA. Chapter 4 focuses on various aspects of AWS security, their challenges, and how ZTA principles can be applied to mitigate these challenges. It also provides an analysis of the various solutions offered by different vendors to realize ZTA in AWS. Chapter 5 demonstrates the POC implementation of a zero trust system to securely access and manage AWS resources such as RDS and EC2 instances. It also provides an evaluation of the POC

[1]Sikt is the Norwegian Infrastructure Provider for Research and Education in Norway. This thesis has been conducted in collaboration with Sikt.

implementation and discusses how this POC can be adopted by organizations like Sikt. Chapter 6 offers a conclusion and relates the work presented in this thesis with the United Nations' (UN) Sustainable Development Goals (SDGs).

# Chapter 2
# Background

This chapter provides background information on different types of cloud and the services offerings by AWS. It discusses traditional security models and their shortcomings. It introduces tenets of ZTA and its logical components and explores different deployment methods of ZTA. Furthermore, it provides real-world implementations of ZTA.

## 2.1 Types of Cloud

In recent years, the landscape of IT infrastructure has undergone a significant transformation. Cloud computing has emerged as a new paradigm, revolutionizing how businesses manage and utilize their IT resources. Traditionally, businesses had to invest in physical hardware and infrastructure to support their operations, often resulting in over-provisioning or under-utilizing resources. With the advent of cloud computing, organizations can scale their resources up or down based on demand, allowing for greater flexibility and cost efficiency. This has empowered businesses with scalability, flexibility, and access to better technologies to manage their fluctuating workloads without the need for significant upfront investment [11].

Cloud computing can be categorized into different types: public, private, community, hybrid, virtual private, and multi-cloud. In the following paragraphs, we discuss different types of cloud [12].

1. **Public cloud** services are provided by third-party cloud service providers, typically offered on a pay-as-you-go or subscription basis. Public cloud providers include AWS, Microsoft Azure, and GCP, among others. They own and manage the infrastructure and services shared among multiple customers. Public clouds offer scalability, flexibility, and cost-effectiveness, making them ideal for small to large businesses, startups, and individuals.

2. **Private cloud** refers to cloud computing resources exclusively used by a single organization or entity. Unlike public clouds, private clouds are deployed within a company's data centers. Private clouds offer greater control, customization, and security than public clouds, making them suitable for organizations with strict regulatory compliance requirements, sensitive data, or specialized workloads. However, private clouds may require higher upfront investment and ongoing maintenance than public clouds.

3. **Community cloud** is a cloud infrastructure shared by several organizations with similar requirements, such as industry-specific regulations, compliance requirements, or security standards. It provides a collaborative platform for multiple organizations to share resources, applications, and services while maintaining isolation and privacy. Community clouds are often deployed and managed by a third-party provider or consortium of organizations that serve a specific community or industry vertical. By pooling resources and expertise, community clouds enable organizations to achieve economies of scale, streamline collaboration, and address common challenges and opportunities within their community.

4. **Hybrid cloud** combines interconnected public and private cloud environments that share data and applications. It allows organizations to leverage the benefits of both public and private clouds while addressing their specific needs and preferences. Hybrid cloud architectures enable seamless data integration, workload portability, and resource optimization across multiple environments.

5. **Virtual private cloud** is an arrangement where the cloud provider allocates a set of virtual resources and provides access to the underlying infrastructure to an organization exclusively. While the cloud provider owns and maintains the cloud infrastructure, the customer retains full control over the allocated virtual resources.

6. **Multi-cloud** refers to using multiple cloud service providers to meet different business needs or leverage specific services and capabilities each provider offers. Organizations may adopt a multi-cloud strategy to avoid vendor lock-in, enhance redundancy and resilience, optimize costs, and access a broader range of services and geographic regions. By distributing workloads and data across multiple clouds, organizations can mitigate risks associated with downtime, data loss, or service disruptions and maintain flexibility to adapt to changing business requirements. However, managing and orchestrating workloads across multiple cloud environments may pose interoperability, security, and governance challenges.

## 2.2 Services Offered by AWS

AWS has emerged as one of the key cloud providers over the last few years. It expands across 245 countries and territories [13], including most European countries, where it plans to provide AWS European Sovereign Cloud to meet the European Union (EU) data privacy laws. AWS offers three kinds of services [2]:

1. **IaaS**: This category offers virtualized computing resources such as virtual machines, storage, and networking infrastructure. The users have full control over the operating systems, applications, and development frameworks running on the provided infrastructure. These include EC2 instances, Simple Storage Service (S3), and RDS, among others.
2. **PaaS**: This model provides a platform to customers so they can develop, run, and manage applications without worrying about the underlying infrastructure. It abstracts away the complexity of managing servers, storage, and networking, enabling developers to focus solely on building and deploying applications. These include Elastic Beanstalk, Lambda, and Aurora Serverless, among others.
3. **SaaS**: This model delivers a complete product, mostly full-fledged software applications hosted and managed by the service provider, AWS in this case. These include Amazon WorkMail, Amazon Chime, and Amazon Connect.

As organizations increasingly adopt cloud services such as AWS, ensuring the security of cloud resources becomes imperative. This necessitates the re-evaluation of traditional security approaches. The following section overviews traditional security models and discusses the associated challenges.

## 2.3 Traditional Security Models

This section briefly discusses the two most commonly used traditional security approaches, perimeter-based and trust-based security models, and their shortcomings.

### 2.3.1 Perimeter-Based Security Model

Perimeter-based security entails fortifying the network perimeter to thwart unauthorized access to internal resources. This model operates under the assumption that threats primarily originate externally and that entities within the network perimeter are inherently trustworthy. For instance, firewalls, which examine and filter incoming and outgoing traffic based on specified criteria, are quintessential examples of perimeter-based security solutions [14]. However, perimeter-based security is generally considered weak in the modern IT landscape. In cloud environments like

AWS, the network perimeter is often porous, and the boundary between internal and external networks is blurred. The perimeter-based security model does not effectively adapt to dynamic cloud environments where resources are ephemeral and constantly changing.

### 2.3.2 Trust-Based Security Model

The trust-based security approach involves granting access to resources based on the level of trust associated with an entity's identity. In this approach, users authenticate themselves through traditional means such as passwords or biometrics and by demonstrating their trustworthiness through various factors such as reputation, past behavior, or endorsements from other trusted entities [15]. However, the trust-based model faces challenges in accurately assessing and quantifying trust levels, leading to inaccuracies in granting access privileges. Additionally, trust is subjective and context-dependent, complicating establishing universal trust criteria. Relying solely on trust may also neglect other critical security factors, such as authentication vulnerabilities and insider threats.

## 2.4 Zero Trust Architecture (ZTA)

The term "Zero Trust," coined in the 1990s, is a security concept that operates on the principle of "never trust, always verify." ZTA attempts to address the shortcomings of the traditional security models. ZTA uses zero trust principles to deploy specific technologies, processes, and controls. It enforces strict access controls, segments networks, authenticates users and devices, encrypts data, and continuously monitors for threats. It advocates for organizations to verify anything trying to connect to their systems before granting access instead of relying solely on traditional perimeter-based and trust-based security measures [6]. This section provides an overview of the tenets and logical components of ZTA. The tenets cover the core ideas behind ZTA, while the logical components describe the essential elements and interactions that constitute ZTA. It also explores various deployment models for ZTA and presents three real-world implementations to illustrate the effectiveness of ZTA.

### 2.4.1 Tenets of ZTA

There are seven tenets of ZTA as defined by National Institute of Standards and Technology (NIST). The tenets are ideal, and the only relevant ones should be implemented according to the needs of the enterprise [6]. They are listed below:

1. Resources: This comprises all the data sources and computing services.

2. Secure Communication: All communications should be secure irrespective of the network location, inside or outside the enterprise network.

3. Session Access: The access to resources is given on a per-session basis with least privilege access, i.e., the authorization and authentication for one resource does not grant access to another.

4. Access Control: The access to resources is determined by dynamic policies based on client identity, application/service, requesting asset state, and other behavioral and environmental attributes, with policies tailored to business needs and risk levels, applying least privilege principles to restrict access.

5. Integrity and Security Posture: The enterprise should evaluate the security posture of all the assets before providing access to the resource. It should continuously monitor and patch devices and applications to make sure they are in their most secure state.

6. Continuous Authentication and Authorization: The authentication and authorization are dynamic and strictly enforced for all the resources. An organization should have identity, credentials, access, and asset management to implement ZTA, which includes using Multi-Factor Authentication (MFA), continuous monitoring with possible re-authentication, and re-authorization.

7. Information Logging: The organization should collect all the information about the current state of the network infrastructure, communications, and assets and use it to improve the overall security posture. This data can also be used for context in access control.

### 2.4.2 Logical Components of ZTA

The ZTA has several logical components that can be operated onsite or offsite via cloud. The subject, resource, Policy Enforcement Point (PEP), Policy Administrator (PA), and Policy Engine (PE) as shown in the Figure 2.1, are the key logical components of ZTA [6]. In a zero-trust access model, a subject requests access to the resource through the PEP. Access is granted based on the decision from its Policy Decision Point (PDP). The functionalities of the three logical components controlling the access are:

- **PE**: It is the system's decision-maker, employing a trust algorithm to analyze enterprise policies and external inputs for access decisions.
- **PA**: It collaborates with the PE to permit or deny access, managing the communication pathway and authentication tokens for client access.

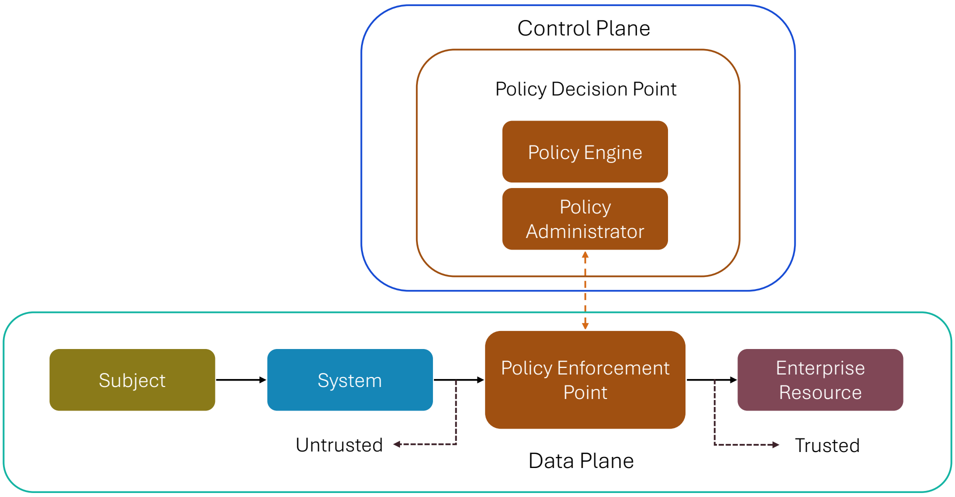



**Figure 2.1:** Zero-Trust Logical Access Model

- **PEP**: It facilitates, monitors, and terminates connections between subjects and resources, encompassing client and resource components, with the trust zone lying on the other side of the PEP.

Apart from the logical components mentioned earlier, various external components as described in [6] contribute to implementing a zero trust system, including continuous diagnostics and mitigation, data access policies, identity management, Security Information and Event Management (SIEM), activity logs, etc.

### 2.4.3 Deployment Models

Different deployment models exist for ZTA, and depending on the need of an enterprise, one of the following models can be selected [6]:

- **Device Agent/Gateway-Based Deployment**: In this model, the PEP is split into two components, with a device agent managing connections on enterprise-issued assets and a gateway acting as a resource proxy. When a subject on an enterprise device seeks access to a resource, the local agent forwards the request to the PA, which evaluates it through the PE. If authorized, a communication channel is established between the device agent and resource gateway, enabling encrypted data flow. This model suits enterprises with robust device management programs, discrete resources, or a preference against Bring-Your-Own-Device (BYOD) policies. It resembles the client-server implementation of the Cloud Security Alliance's Software Defined Perimeter (CSA-SDP) for cloud-heavy enterprises.

- **Enclave-Based Deployment**: In this model, the gateway components are situated at the boundary of a resource enclave, such as an on-location data center. Resources in this model typically serve a single business function or may lack direct communication capabilities with a gateway, as seen in legacy database systems without API support. It's suitable for enterprises using cloud-based microservices for specific business processes, with the entire private cloud placed behind a gateway. This model may also incorporate elements of the device agent/gateway model, where enterprise assets utilize a device agent to connect to enclave gateways, following the same connection process. It's beneficial for enterprises with legacy applications or on-premises data centers that cannot accommodate individual gateways, requiring robust asset and configuration management programs for device agent configuration. However, a downside is that the gateway protects a collection of resources, potentially exposing subjects to resources they lack the privilege to access.

- **Resource Portal-Based Deployment**: In this deployment model, the PEP functions as a single gateway component handling subject requests, either for individual resources or secure enclaves serving specific business functions, such as a private cloud or data center containing legacy applications. Unlike other models, there is no need to install software components on client devices, making it more flexible for BYOD policies and inter-organizational collaborations. However, limited information can be gathered from devices requesting access, and continuous monitoring for threats may be challenging. Without a local agent, the enterprise may lack full visibility and control over assets, mitigated by browser isolation. Yet, assets may be invisible between sessions, exposing the portal to potential attacks such as Denial-of-Service (DoS), necessitating robust provisioning for availability against network disruptions.

- **Device Application Sandboxing**: This is a variation of the device agent/gateway deployment model in which vetted applications or processes run compartmentalized on assets, such as VMs or containers, to shield them from potentially compromised hosts or other applications. Approved applications operate in a sandbox on the subject device, communicating with the PEP for resource access, while requests from other applications are denied. The PEP could be either an enterprise local service or a cloud service. The main advantage is the segmentation of individual applications from the rest of the asset, protecting them from potential malware infections on the host. However, maintaining sandboxed applications for all assets and ensuring security requires additional effort.

### 2.4.4 Real-World Implementations of ZTA

This section explores three prominent real-world implementations of ZTA: Google's BeyondCorp, VMware NSX, and Forrester's Zero Trust eXtended (ZTX) framework. These implementations illustrate that ZTA is effective and essential in modern cybersecurity strategies. The frameworks illustrate how ZTA enhances security, improves operational efficiency, and supports agile business operations. These successful implementations further validate ZTA's effectiveness in protecting against evolving threats and highlight its transformative impact on organizational security.

#### Google's BeyondCorp

Google's BeyondCorp [16, 17] is one of the first implementations of ZTA in an enterprise. It was developed in response to a hacking attempt where the attackers targeted the sources of intellectual property by infiltrating the internal network and utilizing the compromised system for lateral movement. Consequently, Google's primary objective was to diminish the implicit trust that users and devices had developed due to their physical or electronic presence (or both) within the corporate network.

In Google's BeyondCorp framework, only managed devices can access corporate resources. A device inventory database tracks the states and relevant information of all managed devices, while digital certificates and encryption keys are stored in Trusted Platform Module (TPM) or qualified applications for security. User identification requires MFA, conducted via a managed device. To eliminate implicit trust in the legacy network, some managed devices within Google's private network are moved to an unprivileged network and treated like external devices. Authenticated managed devices are assigned to a different virtual network using Virtual Local Area Network (VLAN), while unrecognized devices are placed in a guest or quarantine network.

BeyondCorp employs an internet-facing access proxy akin to a PEP as mentioned in Section 2.4.2 to externalize workflows. Applications are registered with public Domain Name System (DNS), directing requests to the access proxy. After authentication, the proxy consults the Access Control Engine (ACE) to verify and authorize or deny the requests. Central to BeyondCorp, the ACE integrates multiple information sources to determine if a user and device are trustworthy and authorized to access the requested resources. Trust inference continuously updates the trust level of all subjects. Access is granted only if all conditions are met, and the user's trust level exceeds the required minimum for the resource.

BeyondCorp eliminates the traditional Virtual Private Network (VPN)-based access model, allowing employees to work securely from any location without relying on the corporate network perimeter. Each device accessing the network is authenticated and authorized based on its security posture, which includes factors like patch levels, device configurations, and security policies. Users are authenticated through robust identity verification mechanisms, and access decisions are made dynamically based on contextual information, such as user identity, device health, and real-time threat intelligence.

**VMware NSX**

VMware NSX is a network virtualization and security platform that extends zero-trust principles to virtualized environments. It provides granular security controls to protect workloads in data centers and cloud environments. The NSX-based method relies heavily on two key elements: the virtual server and the virtual desktop. NSX employs micro-segmentation, effectively implementing and enforcing policy decisions and trust management across the entire network and at multiple junctures. Consequently, the administrative team can implement precise access control measures across the various granular segments [18].

In the NSX's segmentation of network traffic, security policies are enforced at the VMs level, providing consistent protection regardless of the underlying physical network topology. This helps to ensure that only authorized traffic is allowed between different segments within the data center. Policies are dynamically applied based on factors such as VMs attributes, user roles, and application context. NSX allows integrating third-party security services, such as Intrusion Detection Systems (IDS) and antivirus solutions, into the network flow [19].

Furthermore, VMware NSX provides enhanced visibility and control over east-west traffic within the data center, reducing the risk of lateral movement by attackers. The simplified network management through automation and centralized control reduces the operational burden on IT teams. It enables rapid deployment of new applications and services with consistent security policies, supporting agile business operations.

**Forrester's ZTX**

Forrester's ZTX [20] framework provides a comprehensive approach to implementing zero-trust principles, emphasizing the need for a holistic strategy encompassing networks, devices, applications, and data. The ZTX framework is built around several key pillars, including network security, data security, workload security, device security, identity security, visibility, and analytics.

A centralized segmentation engine orchestrates the division of the enterprise network into multiple micro-core and perimeter segments. This allows for enforcing traffic regulations within the intervals separating these segments, which aligns with the Resource Portal-Based deployment mentioned in Section 2.4.3. However, this approach establishes trust by leveraging information within data packets, which is less precise than solutions deeply integrated with endpoints and services. Moreover, it cannot directly authenticate users with the ZTX segmentation engine, meaning regulations cannot be implemented based on user and device contextual information.

Zero Trust Network Access (ZTNA) builds on the ZTX's concept with several enhancements. ZTNA [21] verifies the credentials of a (user, device) pair to confirm authorization to access an application, establishes a secure (encrypted) connection, and logs and monitors the activity. ZTNA can seamlessly integrate with a software-defined wide area network, secure access service edge, and cloud-based security to ensure consistent protection. It is designed to operate with any transport method or network, including broadband, LTE, and 5G. ZTNA solutions provide secure, granular access to applications and services based on user identity and contextual factors. Data-centric security measures protect sensitive information at rest, in transit, and during processing. By adopting a unified, security-driven networking strategy, institutions can automatically adapt to changes and expansions without compromising access or security.

This chapter (Chapter 2) has provided background information on ZTA, including its tenets, logical components, and different deployment methods. The real-world implementations have demonstrated that ZTA is better than traditional perimeter-based models. The next chapter (Chapter 3) will offer a comprehensive review of the state-of-the-art techniques for implementing the tenets of ZTA discussed in this chapter.

# Chapter 3

# ZTA: Implementation Techniques and Challenges

As discussed in the previous chapter, ZTA has emerged as a significant paradigm shift in the security landscape. It challenges traditional assumptions about network perimeters and trust relationships, advocating for continuous verification and strict access controls. As organizations increasingly rely on cloud services, understanding and implementing zero trust principles becomes paramount to mitigate evolving cyber threats.

In the previous chapter (Chapter 2), Section 2.4.1 describes the tenets of ZTA as defined by NIST. This chapter reviews the state-of-the-art development of techniques central to implementing ZTA tenets. This chapter qualitatively evaluates the ZTA and discusses the challenges associated with implementing ZTA.

## 3.1 Techniques to Implement ZTA Tenets

Table 3.1 presents an overview of the techniques that can be used to implement ZTA tenets. These techniques can be broadly categorized as authentication, access control, encryption, network segmentation, automation and orchestration, and monitoring and analytics (Figure 3.1). Each of these techniques involves multiple levels of granularity. This section provides an in-depth review of the state-of-the-art techniques for implementing ZTA tenets, considering various levels of granularity. Furthermore, the gaps in the existing review literature are presented at the end of each subsection (Table 3.2 to 3.5). Finally, Table 3.6 summarizes the literature reviewed in this section.

### 3.1.1 Authentication

Authentication is a process that verifies the identity of a user, device, or system attempting to access resources within the network. Given the rise in cyber security risks, simple one-time username-password-based authentication is no longer sufficient. Consequently, in a ZTA, authentication is neither a one-time event nor conducted

**Table 3.1:** Techniques to Implement ZTA Tenets

| **ZTA Tenets** | **Techniques for Implementation** |
|---|---|
| Resources | This involves managing users and devices with appropriate access control. |
| Secure Communication | This necessitates encrypting the data-in-transit and segmenting the networks to provide better policy enforcement. All the requests must be continuously authenticated over the network. |
| Session Access | Continuous authentication should be performed to grant access to each session, and fine-grained access control should be implemented based on the principle of least privilege. |
| Access Control | Dynamic and risk-based access control policies, along with contextual and adaptive authentication, can help ensure more secure access to resources. |
| Integrity and Security Posture | All devices should undergo continuous authentication to ensure their safety. Monitoring all devices and their access policies helps to keep a check on compromised devices. |
| Continuous Authentication and Authorization | Authentication should be continuously enforced with MFA, and policies should be applied based on user, resource, and device attributes. |
| Information Logging | Monitoring and providing analytics of ongoing processes help enhance visibility in network infrastructure and access requests. Automating these processes reduces susceptibility to errors, further improving overall security posture. |

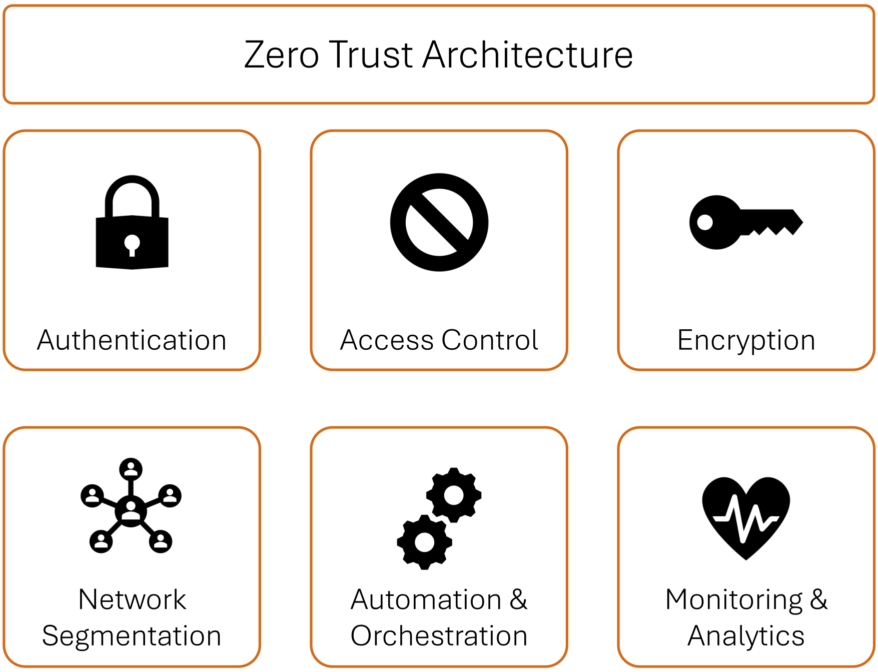



**Figure 3.1:** Techniques to Implement ZTA Tenets

through a simple username-password scheme; rather, it requires MFA and can occur multiple times throughout a session.

Various MFA and continuous authentication mechanisms have been proposed to prevent unauthorized access even after the initial authentication. If a user's behavior during a session appears suspicious, the system may prompt the user to re-authenticate. This approach helps detect and prevent potential security breaches in real time. The following subsections review different authentication methods applicable in a ZTA.

#### Multi-Factor Authentication (MFA)

MFA requires users to provide at least two forms of evidence (factors) to verify their identity [22]. A common form of MFA is Two-Factor Authentication (2FA). These factors could be something the users know (like a password), something users have (like a hardware token [23, 24]), or something users are (like a fingerprint [25]). The use of multiple factors makes it challenging for an attacker to gain access even if they manage to compromise one form of credentials.

The work [26] presents a privacy-preserving MFA using blockchain technology. The system verifies knowledge of a One-Time Password (OTP) with blockchain as a privacy mechanism, ensuring proof of authenticity and confirming the prover's identity. They demonstrate a reliable MFA framework using non-transferable, Non-Fungible Tokens (NFTs) as authentication tokens for identity verification in the final stage of the MFA process.

### Continuous Authentication

User authentication in ZTA faces challenges like continuous verification and the need for specialized equipment. Knowledge-based methods, such as passwords and tokens, only verify users at the moment of access without ongoing tracking.

Continuous biometric authentication is essential for reliable identity verification in ZTA. However, biometric methods require specialized equipment, complicating continuous authentication [27]. Furthermore, traditional features like fingerprints and irises are easily replicable, so continuous authentication must rely on features like electrocardiograms that offer universality, uniqueness, and uninterruptedness.

Shah et al. [28, 29] introduce a Lightweight Continuous Device-to-Device Authentication (LCDA) protocol. It leverages communication channel properties and a configurable mathematical function to dynamically generate session keys for continuous Device-to-Device (D2D) authentication. The work demonstrates that the protocol is resilient to known attack vectors and can easily be deployed in secure D2D communication scenarios, particularly in critical and resource-limited environments.

Meng et al. [30] propose a method to eradicate the need for a trusted node in D2D continuous authentication using blockchain. The devices are divided into three categories: trusted, suspected, and untrusted, and authentication is stopped if any entity is an untrusted device. They demonstrate that achieving a balance between security and efficiency necessitated the adoption of different parameter lengths and intervals tailored for trusted and suspected devices.

The article [31] conducts a comparative analysis of prevalent continuous authentication methods, assessing their suitability for integration within ZTA frameworks. The authors infer that most continuous authentication methods are not universally applicable and are limited to specific input modalities. However, hybrid methods that amalgamate multiple approaches emerge as a promising authentication strategy within the ZTA paradigm.

#### Contextual Authentication

Contextual authentication is another way of authenticating users in a ZTA. It considers contextual information during the authentication process, including the user's location, time of access, type of device used, and network connection. The system might require additional authentication steps or deny access based on this context. For instance, a login request from an unusual location or at any odd time might trigger extra verification.

Contextual authentication tracks changes in physiological and behavioral features over time, including pre-authentication behavior, to continuously verify a user's identity. This method provides a dynamic and adaptive layer of security, as it can detect anomalies or changes in typical user behavior, reducing the risk of unauthorized access. Multimodal biometric authentication enhances security by combining various biometric methods, such as fingerprint, facial, and voice recognition, to create a more robust and reliable authentication system. Similar to MFA, multimodal biometric authentication can leverage multiple biometric traits to ensure that the authentication process is resilient against spoofing and other security threats [27, 32].

Hayashi et al. [33] propose Context-Aware Scalable Authentication (CASA) to balance security and user-friendliness in authentication processes. The framework is designed to dynamically choose an active authentication method that meets specified security criteria considering the passive factors. The authors showcase that location data is one of the most promising passive factors and can always reduce the need for active authentication methods. This reduces the user burden by dynamically adjusting the authentication process.

Benzekki et al. [34] propose a lightweight and cost-efficient context-aware authentication system that only transmits environmental information when a user initiates an authentication request. While the authentication process entails multiple steps to ascertain access, which could potentially detract from user-friendliness, the study demonstrates that the system remains user-friendly despite this delay. However, it's worth noting that the sample size in the study was small.

Ashibani et al. [35] introduce a continuous authentication framework that leverages contextual information for user authentication in smart Internet of Things (IoT) environments. The authors demonstrate that the framework safeguards IoT devices against unauthorized access, locally or remotely, without imposing additional user intervention. It shows that static authentication can be enhanced using contextual information and increased security by providing continuous authentication.

**Table 3.2:** Coverage of Authentication in Existing Review Literature

| Authentication | | | | |
|---|---|---|---|---|
| **Ref.** | **MFA** | **Continuous** | **Contextual** | **Adaptive** |
| [37] | ✓ | | ✓ | |
| [38] | ✓ | ✓ | | |
| [39] | ✓ | | | |
| [40] | | ✓ | | ✓ |
| [41] | ✓ | ✓ | | |
| [42] | ✓ | ✓ | ✓ | |
| [43] | ✓ | ✓ | ✓ | |
| [44] | ✓ | ✓ | ✓ | |
| [45] | ✓ | ✓ | ✓ | |
| [46] | ✓ | | | |
| [47] | ✓ | ✓ | ✓ | |
| [48] | ✓ | ✓ | ✓ | |

### Adaptive Authentication

Adaptive authentication dynamically adjusts the authentication requirements based on the perceived risk level. For example, users may undergo additional authentication steps if they attempt to access sensitive data or perform high-risk operations. This approach aligns with the principle of least privilege, ensuring that users only gain access when necessary and justified.

The work [36] introduces a Unified Authentication Platform (UAP), which generates patterns and evaluates trust to integrate adaptive control. The system evaluates parameters to construct a behavioral profile; any deviation from this is considered a potential risk. This prompts the system to request additional verification steps from the user. In essence, adaptive authentication dynamically adjusts its security measures based on the user's behavior, providing a seamless yet robust authentication experience.

However, there is no one-size-fits-all solution when it comes to security. ZTA treats every access request as a potential threat and verifies the identity of users and devices rigorously and continuously. These robust authentication approaches significantly enhance the security posture of an organization's network. Furthermore, organizations can choose any authentication method that aligns with their specific use case, as discussed above.

Table 3.2 highlights the research gap in the existing review literature for authentication, including MFA, continuous, contextual, and adaptive authentication. It indicates that the majority of the review literature focuses on MFA and continuous authentication, with only a small portion addressing contextual authentication and very few covering adaptive authentication. This lack of research on adaptive authentication can be attributed to the fact that the existing adaptive authentication systems lack methodological design approaches and often face cross-dimensional usability. Consequently, it results in a lack of comprehensive analysis of the attack surface of adaptive authentication systems and its ramifications, considering the diversity of authenticators, contexts, and selection algorithms, all contributing to the complexity of understanding potential vulnerabilities [49].

### 3.1.2 Access Control

Access control lies at the core of ZTA as the primary mechanism for enforcing the principle of least privilege and continuous verification of user identities and device integrity. As organizations navigate the complexities of securing their cloud infrastructure deployments, implementing robust access controls becomes imperative to safeguard sensitive data and critical assets from ever-evolving threats. Access to a resource is given only when the subject satisfies the access policy, which is maintained as an integral part of the access control mechanism formulated based on the organization's use case. These policies can be defined and configured in several ways. In this section, we review different types of access control policies.

#### Role-Based Access Control (RBAC)

RBAC assigns permissions to a single user or a group of users based on their roles within an organization. Users are grouped into roles based on their job functions or responsibilities, and permissions are assigned to these roles. In an organization, roles like "Administrator," "Manager," and "Employee" may be defined, each with its own set of permissions. Users are assigned to these roles, and their access rights are determined by the permissions associated with their roles.

The three primary steps for implementing RBAC are role assignment, role authorization, and permission authorization. Different RBAC models have been proposed based on these steps: flat, hierarchical, constrained, symmetric, and temporal. The flat RBAC model incorporates the three primary steps of RBAC, allowing for many-to-many user-to-role and permission-to-role assignments. The hierarchical RBAC model builds on the flat model by defining seniority relationships, where senior roles encompass all permissions of the roles below them. The constrained RBAC model extends the hierarchical model by introducing separation of duties, requiring multiple individuals to complete certain tasks. Symmetric RBAC, the most advanced form, includes all features of constrained RBAC and adds support for permission-role review. Lastly, the temporal RBAC model further extends RBAC by allowing the enabling and disabling of roles based on time constraints [50].

A Task-Role-Based Access Control (T-RBAC) model was introduced in the work [51], proposing a task-oriented concept. A task-oriented concept focuses on associating permissions with specific tasks rather than general roles. This model's primary objective is creating a secure workflow system access mechanism. Despite its utility within distributed systems, the T-RBAC model falls short in complex cloud network environments. It neglects the issue of user rights assignment, merely introducing a set of trustees to act as task executors without exploring the process of trustee determination in a real-world scenario.

Addressing the trust oversight identified in T-RBAC, [52] presents a viable solution to address the trust concerns. The authors propose an integrated approach that combines a comprehensive trust model with cryptographic T-RBAC, aimed at enhancing the privacy and security of data stored in cloud systems. In the proposed design, the risk associated with cloud storage systems can be mitigated, thereby enhancing overall security. Additionally, this approach can potentially enhance the decision-making capabilities of cloud operators and data owners, thereby improving the overall quality of service.

In the work [53], the authors emphasize the separation of policy enforcement (PEP) from decision-making (PDP) through RBAC. By separating enforcement from decision-making, RBAC can enable a more flexible and adaptable approach to access control, which is well-suited to the dynamic and continuously evolving nature of zero trust environments.

The work [54] integrated the RBAC model with trust management principles, unveiling a novel Distributed Role-Based Access Control (dRBAC) model tailored for distributed environments. Within this framework, roles are endowed with access privileges and empowered to delegate permissions to other roles. It uses continuous monitoring to ensure the ongoing validity of established trust relationships throughout extended interactions.

Jung et al. [55] introduced an adaptive resource access control framework based on the RBAC model. This improved model can dynamically adjust resource security levels, effectively addressing the challenge of fluctuating environmental variables within cloud computing environments.

Xu et al. [56] proposed an RBAC model for cloud environments, specifically cloud storage, combining an identity-based cryptosystem. The proposed solution aims to augment RBAC's efficiency and flexibility. This is achieved by incorporating a mechanism for role inheritance, which optimizes the assignment of permissions, making it both efficient and precise. The model was subjected to functional testing and performance analysis, demonstrating its ability to perform various functions, understand the dynamic access control of ciphertext data, and maintain an acceptable operation completion time.

RBAC enhances security by assigning distinct access levels based on user roles, thereby improving confidentiality and privacy. It facilitates duty segregation and provides flexibility. However, overly granular permissions can lead to complexity and management challenges, known as 'role explosion.' Implementing RBAC requires a deep understanding of the organization's security landscape and existing permissions.

While RBAC might be straightforward during initial implementation, defining roles may become more intricate as new roles and users are introduced. Expanding an RBAC within businesses can be costly and intricate, necessitating concurrent scaling of infrastructure contingent upon the organization's scale and complexity. Furthermore, RBAC's scope is often constrained by its focus solely on employee roles for authorization, leading to limitations in scalability and dynamism.

### Attribute-Based Access Control (ABAC)

ABAC dynamically evaluates access requests based on attributes associated with the user, resource, and environmental context. Attributes can include user attributes (e.g., role, department), resource attributes (e.g., sensitivity level, location), and contextual attributes (e.g., time of access, device posture). It provides finer granularity and is more scalable than other access control models [57].

In ABAC, access decisions are made based on the combination of attributes. For instance, access to a confidential document may be granted only if the user's role is "Manager," the document's sensitivity level is "Confidential," and the access request originates from within the corporate network. ABAC empowers organizations to regulate resource access through intricate policies and conditions, rendering it exceptionally adept in dynamic and intricate environments.

Unlike RBAC, where administrators can easily view assigned user privileges, ABAC lacks a straightforward method to look up users' access permissions. As a result, verifying access requires checking each object against the access policy. Additionally, configuring numerous policies to determine user attributes for resource access can be time-consuming due to the complexity of the technology [58]. Hybrid ABAC models and frameworks have been explored to address the challenges and limitations of ABAC.

The work [59] proposed a Location-Aware Attribute-Based Access Control (L-ABAC) for secure data access control in cloud environments. This model involves various entities such as a data owner, attribute authority, location servers, sensors, data consumers (users), and the cloud server. The advantage of this method is that when compromised, a single server only exposes information linked to a particular location, leaving other data confidential.

García-Teodoro et al. [60] propose a Security Attribute-Based Dynamic Access Control (SADAC) aimed at fortifying security within corporate networks and service provider environments. The model relies on a dynamic policy enforcement point that dynamically grants or denies access to a device to the network environment and an anomaly detection module that accesses the security posture demonstrated by users and devices, enabling dynamic decisions regarding access permissions.

The work [61] proposes an Attribute and User Trust Score-Based Zero Trust Access Control Model (AU-ZTAC) that merges the principles of zero trust and ABAC. It aims to fulfill network protection needs while ensuring fine-grained dynamic access control and integrating trust evaluation into the access control process to capture users' intentions accurately. The work validates the efficacy and viability of this model for security. However, this model can be easily applied for network protection in zero trust environments where resource access is dynamically controlled.

A dynamic access control model is essential to manage access requests from numerous dynamic users. Riad et al. [62] proposed a hierarchical access control scheme utilizing dynamic revocation threshold vectors based on multi-dimensional access control to dynamically authorize or revoke users with multiple rights in the cloud, optimizing encryption and decryption processes.

Yao et al. [9] introduced another way to define access policies dynamically based on contextual information and risk assessments. These policies can adapt to changes in user roles, device status, or environmental conditions, ensuring that access permissions always align with security policies and business requirements. This hierarchical access control model uses partial trust-based access control, where the user behavior is used to generate user trust and portraits.

The work [63] proposes an integrated identity and attribute-based access management system for cloud web services. Combining adaptive authentication with ABAC enhances the security of cloud web services. In this ABAC, user attributes are evaluated before authorization is granted. This approach also eliminates the role mapping requirement in the RBAC mechanism.

The work [64] demonstrates a formal ABAC model using Event-Calculus. The model can accurately represent and verify AWS IAM policies. The expressive and adaptable approach features generic Event-Calculus models and tool support to automatically convert JSON-based IAM policies into Event-Calculus. Performance evaluations conducted on real IAM policies demonstrate the scalability and practicality of this method.

Gupta et al. [65] delve into ABAC within the context of cloud-enabled industrial smart vehicles, facilitating location-specific and real-time notifications in smart transportation networks. Their innovative security solution integrates a finely-tuned ABAC model, incorporating dynamically assigned groups based on the characteristics of the vehicles in motion, and the practicality and usability are demonstrated by deploying a prototype implementation on AWS.

#### Identity-Based Access Control (IBAC)

IBAC is an access control model that determines access permissions based on the identity of the user requesting access. In IBAC, access decisions are made solely based on the user's identity, typically in the form of a username, user ID, or other unique identifier. Unlike RBAC or ABAC, which consider roles, attributes, or contextual factors, IBAC relies solely on the user's identity to determine access rights.

Yang et al. [66] introduce cloud storage auditing protocol using cryptography with IBAC. It allows data owners to control access to their sensitive information, ensuring that only authorized users can access it. It simplifies the implementation process by eliminating the need for complex certificate management.

IBAC provides a straightforward approach to managing access permissions based on user identities. Still, it may not be suitable for complex access control scenarios that require more nuanced control over access rights. It also faces scalability challenges, especially in environments characterized by dynamically shifting groups of subjects and objects. The need for frequent updates to access authorizations becomes impractical under such circumstances.

#### Function-Based Access Control (FBAC)

FBAC grants access based on the functions or operations users can perform. Rather than assigning permissions to roles or attributes, access is determined by the specific functions or operations users are authorized to execute. In FBAC, permissions are associated directly with functions or operations within the system. For example, a user may be granted permission to "Create User Accounts" or "Approve Purchase Orders." Access is granted based on whether the user can perform these specific functions.

In the work [67], authors propose FBAC based on the functional encryption. Unlike traditional methods where access authorizations are stored in a two-dimensional Access Control Matrix, FBAC employs a three-dimensional tensor, termed an access control tensor, to manage access rights. This approach enables applications to implement access controls with unprecedented granularity, allowing users to invoke authorized commands on specific data segments. This level of customization is not possible with the classical access control matrices.

While FBAC has not yet been explored in depth, its deployment could enhance access control capabilities, especially in the emerging IoT domain.

### Risk-Based Access Control (RiBAC)

RiBAC is a dynamic access control approach that grants or denies access to resources based on evaluating risk factors associated with the user, the resource accessed, and the current context. For example, a user attempting to access sensitive data from an unfamiliar location or using an unsecured device may be granted limited access or subjected to additional authentication measures to mitigate the elevated risk.

The authors in [68] analyze and evaluate the risk factors present in the risk-based access control model and infer that a significant challenge is developing an accurate and efficient risk estimation technique applicable across various domains.

Vanickis et al. [8] introduce a policy enforcement framework called Fuzzy Risk Framework for Zero Trust Network (FURZE) [69] to tackle open challenges for RiBAC within Zero Trust Network (ZTN). The authors detail the design of required policy languages to facilitate fuzzy risk evaluation and to implement policy enforcement for RiBAC.

### Blockchain-Based Access Control

Blockchain-based access control is a security mechanism that uses blockchain technology to manage and verify access permissions in a decentralized and tamper-proof manner. Yang et al. [70] propose AuthPrivacyChain, a blockchain-based access control framework with privacy protection. In this framework, they utilize the blockchain node's account address as an identity, redefining data access control permissions for the cloud. These permissions are encrypted and stored on the blockchain. They demonstrate that AuthPrivacyChain effectively prevents unauthorized access by hackers and administrators while preserving authorized privacy.

Blockchain-based access control has also been integrated with other access control models such as RBAC and ABAC. The work [71] proposes a blockchain-based access control to facilitate lightweight data sharing among diverse organizations, particularly for zero-trust environments. In this approach, a RBAC model is deployed using a unique multi-signature protocol and smart contract methods. This innovative approach demonstrates a shift from conventional access control schemes, which often

**Table 3.3:** Coverage of Access Control in Existing Review Literature

| Access Control | | | | | | |
|---|---|---|---|---|---|---|
| **Ref.** | **RBAC** | **ABAC** | **IBAC** | **FBAC** | **RiBAC** | **Blockchain** |
| [37] | | | ✓ | | | ✓ |
| [38] | | | | | ✓ | |
| [39] | ✓ | | | | | ✓ |
| [40] | | | | | ✓ | |
| [41] | | | | | | |
| [42] | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| [43] | ✓ | ✓ | ✓ | | | |
| [44] | ✓ | ✓ | ✓ | | ✓ | |
| [45] | ✓ | ✓ | | | ✓ | |
| [46] | ✓ | ✓ | | | | |
| [47] | ✓ | ✓ | | | ✓ | |
| [48] | ✓ | ✓ | | | | ✓ |

focus on protecting a single entity, acknowledging the need for multi-party protection despite the potential for increased security costs.

Ding et al. [72] introduced ABAC model leveraging blockchain technology. They demonstrated that it can be efficiently implemented on IoT systems and withstand multiple attacks compared to the traditional access control technologies unsuitable for direct implementation in IoT systems due to their complex access management and centralized nature, compromising credibility. This model offers a decentralized and scalable access control system, requiring less trust and enhancing the existing system's robustness.

Awan et al. [73] propose a novel secure framework, Zero Trust and Attribute-Based Access Control for Internet of Things using Blockchain (ZAIB) for IoT-based smart environments. This framework employs ZTA and ABAC to monitor device-to-device communications, facilitating different access control levels based on environmental parameters and device behavior. Utilizing blockchain for anonymous registrations and immutable logs and Interplanetary File System (IPFS) for protecting attributes, trust level histories, and data, ZAIB offers robust security measures. The evaluation demonstrates that ZAIB effectively meets active defense and end-to-end security enforcement requirements. This approach overcomes the problem of resource limitations and the dynamic nature of the IoT network.

Table 3.3 highlights the research gap in the existing review literature for access control. The types of access control discussed include RBAC, ABAC, IBAC, FBAC, RiBAC, and blockchain-based access control. The table shows that most review literature focuses on RBAC and ABAC, with only some review papers focusing on the other types. This can be attributed to the fact that RBAC is easier to implement and ABAC is more scalable than other models, making them more prominent in research. The remaining access control models receive less attention as some are challenging to implement, and some are relatively new.

### 3.1.3 Encryption

Encryption converts data into an unreadable format that can only be accessed with the correct decryption key, ensuring its confidentiality from unauthorized access and maintaining its integrity against tampering. In ZTA, end-to-end encryption is a fundamental principle covering data-at-rest and data-in-transit. All data stored within the system, such as servers, databases, and storage devices, must be encrypted. Similarly, all data transmitted over networks, whether internal or external, should be encrypted to prevent unauthorized access and tampering.

Research by Weever and Andreou [74] underscored the importance of securing data during transit between containerized applications. Their findings highlighted vulnerabilities in transit data, susceptible to attacks leading to data breaches. Therefore, implementing robust encryption protocols and securely managing encryption keys are critical to maintaining the security posture of a ZTA environment, ensuring that sensitive information remains protected against potential breaches throughout its life cycle.

In this section, we first discuss different encryption mechanisms relevant for a ZTA, and then we discuss different key management strategies proposed in the literature.

#### Encryption protocols

Various cryptographic schemes have been proposed to address specific encryption challenges. For instance, Ciphertext-Policy Attribute-Based Encryption (CP-ABE), introduced in [75], enforces complex access control based on user attributes, ensuring that only users with the correct attributes can decrypt the data.

Helil et al. [76] introduced another CP-ABE, which addresses the challenge of preventing unauthorized successive access to multiple data objects. It provides access control for sensitive datasets in cloud storage, ensuring data confidentiality, collusion resistance, and policy privacy.

Key-Policy Attribute-Based Encryption (KP-ABE), introduced in [77], provides fine-grained sharing of encrypted data stored by third parties on the internet. KP-

ABE addresses the limitations of coarse-grained sharing in traditional encryption methods by allowing selective decryption based on attribute matching without requiring users to share private keys.

Since KP-ABE cannot support ciphertext-policy access control, Li et al. [78] introduced a privacy-aware multi-authority ciphertext-policy Attribute-Based Encryption (ABE) scheme with accountability to enhance user privacy and data security in cloud environments. It identifies dishonest users who share their decryption keys and provides a fine-grained access control system.

Fuzzy Identity-Based Encryption (IBE) introduced by Chatterjee et al. [79] offers error tolerance suitable for biometric-based encryption, accommodating inherent noise in biometric data. This innovation enhances user encryption with specific attribute sets, proving its security against collusion attacks.

Role-Based Encryption (RBE) schemes, like those by Zhou et al. [80] and Sultan et al. [81], enhance secure cloud storage by controlling access based on user roles and preventing cloud providers from reading plain data. These schemes support efficient user revocation without re-encryption and maintain constant ciphertext and key sizes. Sultan et al.'s scheme further allows data sharing across organizations with an outsourced decryption mechanism to reduce user computational load. Security analyses confirm this scheme is secure against chosen plaintext attacks and has low computational overhead, making it practical for ZTA.

Patil et al. [82] demonstrated a blockchain-based consensus algorithm using cryptographic techniques and decentralized ledgers to enforce stringent security protocols, ensuring all data exchanges are secure and verifiable without relying on a central authority. Advancements in searchable encryption schemes provide secure search capabilities over encrypted data. The work in [83] presented a new searchable encryption scheme utilizing the modular inverse to create a probabilistic trapdoor for secure searches over an inverted index table. Such schemes offer lightweight and secure solutions for client-server architectures, suitable for implementations in ZTA.

While encryption is vital for security, balancing it with efficiency is challenging. Encrypting all data can be costly and inflexible, especially with specialized hardware. Symmetric encryption, essential for securing databases and communications, is often overlooked in ZTA discussions. Although patterns for cryptography are addressed in [84, 85], performance studies specific to ZTN are lacking. Models from NIST and other organizations recommend using Public Key Infrastructure (PKI) but do not specify the application of particular cryptographic controls.

Moreover, recent research has focused on enhancing CP-ABE schemes with functionalities like user revocation and ciphertext updates, ensuring both forward (revoked users cannot access past data) and backward (revoked users cannot access future data) security. These schemes offer practical and cost-effective solutions for secure data sharing in ZTA.

Xiong et al. [86] proposed an efficient CP-ABE scheme that uniquely achieves partially hidden policy, direct revocation, and verifiable outsourced decryption. The scheme introduces partially hidden policies to protect private information in access policies. It ensures that revoked users cannot access messages after revocation without affecting non-revoked users.

The work [87] enhances CP-ABE by adding user revocation, secret key delegation, and ciphertext update functionalities, resulting in a Revocable-Storage Hierarchical Attribute-Based Encryption (RS-HABE) scheme. RS-HABE ensures forward security and backward security.

**Key Management**

Managing encryption keys is critical for encryption. Key management is a process that involves the generation, distribution, storage, rotation, and disposal of encryption keys. Secure key management systems are necessary to maintain the security of encrypted data. [88].

Effective key management also encompasses key lifecycle management, which includes key usage monitoring, auditing, and revocation procedures. Monitoring the usage of keys helps detect unauthorized activities or anomalies promptly, ensuring the integrity and confidentiality of encrypted data. Auditing key activities provides a comprehensive view of how keys are accessed and used over time, aiding in compliance with security policies and regulations. Additionally, secure backups of encryption keys are essential to prevent data loss due to accidental deletion or system failures [89].

Li et al. [90] proposed a revocable IBE scheme incorporating outsourcing computation, proposing a server-aided model where a Key Update Cloud Service Provider handles most key generation tasks, reducing the workload for both the key generation service provider and users.

Similarly, Revocable-Storage Identity-Based Encryption (RS-IBE) proposed in [91] and Re-Randomizable Attribute-Based Encryption (RRABE) proposed in [92] ensure forward and backward security, enabling user revocation and ciphertext updates while maintaining decryption key security. This approach ensures the secret key remains secure even if the decryption and key update are compromised.

Implementing robust encryption techniques and effective key management strate-

**Table 3.4:** Coverage of Encryption in Existing Review Literature

| **Coverage of Encryption in Existing Review Literature** | | | | | | |
|---|---|---|---|---|---|---|
| **Ref.** | **Symmetric** | **Asymmetric** | **ABE** | **IBE** | **Data-at-rest** | **Data-in-transit** |
| **[37]** | | | | | | ✓ |
| **[38]** | | | | | | ✓ |
| **[39]** | | | | | | |
| **[40]** | | | | | | ✓ |
| **[41]** | | | | | | |
| **[42]** | ✓ | ✓ | | | ✓ | ✓ |
| **[43]** | | | ✓ | | | |
| **[44]** | | | | | ✓ | ✓ |
| **[45]** | | | | | ✓ | |
| **[46]** | ✓ | ✓ | | | | |
| **[47]** | | | | | ✓ | ✓ |
| **[48]** | | | | | ✓ | |

gies is essential for securing data in ZTAs. Advanced schemes like CP-ABE, KP-ABE, and RRABE provide flexible and resilient solutions for modern data security challenges.

Table 3.4 highlights the research gap in the existing review literature for encryption. It shows that most of the review literature generally recommends encrypting data-at-rest and data-in-transit without specifying the encryption methods or cryptographic techniques to be used.

### 3.1.4 Network Segmentation

Network segmentation divides the network into smaller, isolated networks or segments. This limits the scope of trust and restricts lateral movement within the network. Network segmentation is an essential part of a ZTA, and it can be implemented using different approaches as described by NIST [6]. Micro-segmentation and software-defined segmentation, also known as Software-Defined Perimeter (SDP), are the two most commonly used approaches to implement ZTA in an enterprise's network. This section describes these two network segmentation approaches proposed in the literature.

### Micro-Segmentation

Micro-segmentation's fundamental principle involves implementing security policies near the resource that needs safeguarding, effectively dividing the network infrastructure into smaller logical "segments" to efficiently protect individual resources or their logical groups. By allowing only authorized entities within the data center to access the applications or data on protected resources, micro-segmentation inhibits an attacker's ability to move laterally within the network. Micro-segmentation enhances the network security in the complex and dynamic cloud networks. It reduces lateral movement without affecting the network performance and improves robustness while decreasing the network exposure as shown in [93, 7].

The work [94] explores a network security architecture tailored for cloud computing environments. This architecture supports a zero trust approach by utilizing network micro-segmentation to inspect network traffic based on port and protocol information, allowing only authorized communications. It leverages Illumio's tool to demonstrate micro-segmentation, involving two main components: PE and the Virtual Enforcement Node (VEN). The PE writes and enforces policies based on traffic data, while VENs collect telemetry and enforce these policies. The implementation involved creating secure connections between the virtual machines, demonstrating successful micro-segmentation with no packet loss, indicating no network congestion or collisions.

The paper [95] introduces micro-segmentation to prevent lateral movement within Industrial Internet of Things (IIoT) networks. The automated micro-segmentation model is based on machine learning clustering algorithms, which generate micro-segments based on network traffic and a decision tree classifier to classify traffic as malicious or normal. The security analysis shows that micro-segmentation reduces device infection rate by limiting lateral movement.

The authors in [96] propose a micro-segmentation based solution known as "eZTrust." This helps to enable fine-grained access control policies based on workloads and can be efficiently enforced without relying on the network. eZTrust uses extended Berkeley Packet Filter (eBPF) to trace authentic workload identities and shows lower packet latency and CPU overhead than traditional perimeter-based solutions.

The concept of micro-segmentation is also applied to 5G networks to improve their security [97]. It utilizes Software-Defined Networking (SDN) and Network Function Virtualization (NFV) to create flexible and scalable network architectures. It is adapted for 5G networks to provide fine-grained isolation and tailored security policies for applications and services.

The paper [98] proposes using micro-segmentation, facilitated by the edge cloud, to enhance the security of smart home networks. It significantly reduces the attack surface, especially against threats like the Mirai botnet, while minimally disrupting valid device communications.

### Software-Defined Perimeter (SDP)

The principle of SDP is to confirm and authenticate the client's identity before establishing any communication. This contrasts with traditional networks that permit clients to establish connections (e.g., TCP/IP) before authentication. SDP implementations comprise an SDP controller, which authenticates and authorizes the clients, and an SDP gateway, which connects to the applications. In this setup, application servers remain invisible to the client, and the client's communication with these servers is authorized and facilitated by the controller and the gateway [99].

The work [100] demonstrates SDP in a virtualized network for enterprise scenario, that though the SDP framework increases initial connection setup time, it effectively mitigates DoS and port scanning attacks, confirming its potential as a robust security solution. The SDP architecture includes components like Single Packet Authentication (SPA), mutual Transport Layer Security (TLS), and dynamic firewalls, ensuring secure, encrypted connections and addressing various network security issues. Despite its benefits, SDP faces challenges like potential network disruption, configuration updates, and controller vulnerability.

The work [101] proposes SDP to secure IaaS. The SDP is implemented and tested in the AWS cloud environment, and port scanning is also used to verify its effectiveness. The study also shows that SDP can resist DoS attacks while maintaining access for legitimate users. The findings highlight SDP's potential and pave the way for future research in securing IaaS.

The article [102] proposes the SDP as a framework to manage connections and restrict network access within SDN-enabled infrastructures. Integrating SDP with SDNs offers scalable and managed security solutions, allowing flexible deployment tailored to various network security needs. The framework is tested using virtualized network testbeds, demonstrating its resilience to port scanning and DoS bandwidth attacks. Additionally, the results highlight promising integration points between SDP systems and SDNs, suggesting avenues for further research.

Through these forms of network segmentation, ZTA aims to minimize the potential attack surface, contain the spread of threats, and reduce the potential impact of a security breach. Table 3.5 highlights the research gap in the existing review literature for network segmentation. It shows that most review literature focuses on micro-segmentation while only some focus on SDP.

**Table 3.5:** Coverage of Network Segmentation, Automation and Orchestration, Monitoring and Analytics in Existing Review Literature

| | Network Segmentation | | Automation & Orchestration | | Monitoring & Analytics | |
|---|---|---|---|---|---|---|
| **Ref.** | **Micro Segmentation** | **SDP** | **Automation** | **Orchestration** | **Monitoring** | **Analytics** |
| **[37]** | ✓ | | ✓ | ✓ | ✓ | ✓ |
| **[38]** | | | | | ✓ | |
| **[39]** | ✓ | ✓ | ✓ | | ✓ | ✓ |
| **[40]** | | | ✓ | ✓ | ✓ | |
| **[41]** | ✓ | ✓ | | | ✓ | |
| **[42]** | ✓ | ✓ | ✓ | ✓ | ✓ | |
| **[43]** | | ✓ | | | | |
| **[44]** | ✓ | ✓ | | | ✓ | ✓ |
| **[45]** | ✓ | | ✓ | | ✓ | ✓ |
| **[46]** | ✓ | ✓ | | | ✓ | ✓ |
| **[47]** | ✓ | | ✓ | ✓ | ✓ | ✓ |
| **[48]** | ✓ | | ✓ | ✓ | ✓ | |

### 3.1.5 Automation and Orchestration

In the context of ZTA, automation refers to the use of technology to perform tasks with minimal human intervention, while orchestration involves the coordination and management of these automated tasks to achieve specific goals. Automation and orchestration are fundamental to maintaining a dynamic and proactive security environment that can adapt to evolving threats in a ZTA. They contribute to a robust security posture by enabling real-time responses and continuous validation, which are fundamental to the ZTA. These concepts help streamline and manage a zero-trust network's complex and dynamic environment, ensuring security policies are consistently applied and operational efficiency is maintained.

Incorporating sophisticated automated actions and responses is paramount for organizations embracing zero-trust solutions. The work [103] highlights the necessity of evaluating manual security operations and transitioning them into automated processes to enhance the efficacy of zero-trust capabilities. Within the realm of ZTA, automation extends to decision-making processes by utilizing Artificial Intelligence (AI) [104]. By harnessing acquired intelligence, whether internally or externally sourced, organizations can proactively thwart malicious actions and implement corrective measures promptly. This proactive approach minimizes human error and response duration, bolstering security effectiveness.

While traditional attack detection methods can help ZTA effectively identify internal and external threats, they often fall short in classifying and managing these security events. They also cannot automatically alert the security administrator or initiate countermeasures. Automated SIEM orchestration [105] addresses this issue by automatically collecting and analyzing information from the attack detection system and triggering security alerts to provide remediation or mitigation solutions.

Supervised learning classification algorithms are employed to automatically categorize various security events, thus resolving the inefficiencies of manual classification. These algorithms use labeled datasets to learn the characteristics of different types of security incidents, enabling them to identify and classify new events accurately and rapidly. This automation significantly enhances the efficiency and accuracy of security event management, reducing the response time and mitigating potential threats more effectively.

The cloud is a significant application scenario for ZTA, given that many applications store their data there. However, continuous user interaction with the cloud can lead to data breaches due to the persistent exchange of information across potentially insecure channels. To address these challenges, Esposito [106] proposed orchestrating different access control models to enhance security. This approach integrates multiple access control mechanisms to create a cohesive and robust security framework. Additionally, Esposito introduced a pseudonym-based privacy protection method, which safeguards users' personal information by replacing real identifiers with pseudonyms. This technique ensures that the users' real identities remain protected even if data is intercepted or breached.

Organizations can consistently apply security measures, enhance operational efficiency, and reduce human error by automating and orchestrating security policies, procedures, and responses. Integrating advanced authentication techniques, securing IoT devices, and employing sophisticated access control models for cloud environments further strengthen the security framework. Integrating automation and orchestration within ZTA will remain crucial for maintaining a robust and resilient security posture as threats evolve. Table 3.5 highlights the research gap in the existing review literature for automation and orchestration. It shows that only some review literature focuses on automation and orchestration.

### 3.1.6 Monitoring and Analytics

Monitoring and analytics are critical to the efficacy of a ZTA, providing the visibility and insights needed to enforce security policies, detect threats, and ensure compliance. These components help maintain a comprehensive understanding of network activities and user behaviors, which is vital for a proactive and adaptive security posture.

Threat intelligence involves gathering information about threats to help organizations enhance their security by mining publicly available resources for vulnerabilities, cyber-attacks, and other relevant data. Given the numerous sources of threat intelligence, an automated system is necessary to collect and verify the authenticity of the intelligence.

Sentuna et al. [107] discussed emerging cyber threat intelligence technologies and proposed a threat prediction model that combines the naive Bayes posterior probability function with a risk assessment function. Hacker forums are a crucial data source for cyber threat intelligence. Additionally, deep learning and transfer learning methods have been utilized to improve the performance of threat intelligence identification and reduce model training time [108, 109].

Automated threat intelligence collection, leveraging AI algorithms, has become common for sensing, reasoning, and detecting advanced cyber attacks. Much cyber threat intelligence is sourced from open communities such as forums, blogs, and tweets. AI technologies can automatically extract useful information from these sources, making the process more efficient and effective [110, 111].

Cascavilla et al. [112] reviewed existing cybercriminal activities, enhancing their classification for risk assessment. Their study categorizes cybercrimes and provides insights into cybercriminals' evolving tactics and strategies, helping organizations better understand and mitigate potential risks.

In contrast, Wagner et al. [113] focused on automated cyber threat intelligence sharing techniques. Their research delves into mechanisms and technologies that facilitate the timely and secure exchange of threat intelligence among different entities, such as organizations, government agencies, and cybersecurity vendors. These techniques enhance collective defense capabilities and improve responses to emerging cyber threats by automating the sharing process.

Unsupervised learning clustering algorithms group different threat intelligence patterns based on similarity, while log-based anomaly detection methods use AI for automatic log monitoring and identification. Given the vast amount of log data, often in billions, unsupervised learning methods like dimensionality reduction algorithms can significantly reduce computational costs and improve anomaly identification efficiency [114]. Additionally, deep reinforcement learning is used to collect threat intelligence, where agents actively learn to extract more accurate threat intelligence through trial and error, enhancing identification performance [115].

System logs provide near real-time feedback on the operation of system components, and automated anomaly detection in these logs can quickly identify abnormal access activities. System logs typically record system operations in a time-series format, and supervised learning-based log analysis methods can automate the extraction of time-series anomaly features. Deeplog [116], for instance, focuses on constructing workflows from underlying logs and analyzing detected anomalies. However, Wang and Ji [117] argued that Deeplog's performance is unsatisfactory, and they optimized it by incorporating a first-order outlier detection algorithm, proposing a semi-supervised anomaly detection model.

Automated anomaly detection in log files can promptly identify abnormal or illegal behaviors within ZTA's internal resources. Soldani and Brogi [118] reviewed methods for anomaly detection and the causes of anomalies in cloud services. Landauer et al. [119] explored clustering methods for analyzing large volumes of log data, while Chalapathy and Chawla [120] provided an overview of deep learning-based anomaly detection methods, evaluating their effectiveness.

Semi-supervised learning is primarily used to detect logs with abnormal conditions in ZTA. While effective for identifying known attack behaviors, its performance declines when faced with unknown attacks. Semi-supervised learning leverages large amounts of unlabelled and labeled data to perform pattern recognition, preventing data and resource wastage. This approach addresses the issues of weak generalization in supervised learning models and the inaccuracy of unsupervised learning models [121].

Despite continuous authentication and authorization within the ZTA architecture, incidents like identity fraud can still occur. Continuous monitoring of internal users and devices is an effective solution. AI techniques model user behavior extracted from logs to automate identifying abnormal user activity. Given the scarcity of papers reviewing AI-based user abnormal behavior monitoring, potential AI-based approaches to user behavior detection are discussed. Clustering algorithms identify legitimate users' illegal behaviors by grouping normal user behaviors into clusters, labeling users outside these clusters as abnormal [122, 123].

Deep learning algorithms can learn users' daily access habits and classify behaviors as normal or abnormal. Additionally, deep learning can predict potential user behaviors based on context and historical data, enabling timely detection or prevention of threats. Unsupervised recurrent deep learning methods can be used for anomaly detection in user behavior during interactions with organizational resources [124]. The work [125] proposed a system that monitors database transactions to detect anomalies, uses encryption to protect data during backups, and implements a recovery protocol that ensures quick data restoration with minimal downtime. This combination enhances overall data security and operational efficiency in cloud environments.

In summary, integrating AI-driven monitoring and analytics within ZTA can provide robust tools for enhancing security measures. Organizations can significantly improve their threat detection and response capabilities by leveraging advanced techniques such as unsupervised, deep, and semi-supervised learning. Continuous monitoring of user and device behavior and effective threat intelligence gathering ensures a proactive and adaptive security posture, essential for combating the ever-evolving landscape of cyber threats. Table 3.5 highlights the research gap in the existing review literature for monitoring and analytics. It shows that most of the

**Table 3.6:** Overview of the Review on Techniques to Implement ZTA Tenets

| **Techniques** | **Granularity** | **References** |
|---|---|---|
| **Authentication** | MFA | [22], [25], [26] |
| | Continuous Authentication | [27], [28], [29], [30], [31] |
| | Contextual Authentication | [27], [32], [33], [34], [35] |
| | Adaptive Authentication | [36], [49] |
| **Access Control** | RBAC | [50], [53], [51], [52], [54], [55], [56], [71] |
| | ABAC | [57], [58], [59], [65], [9], [60], [61], [62], [63], [64], [72], [73] |
| | IBAC | [66] |
| | FBAC | [67] |
| | RiBAC | [68], [8], [69] |
| | Blockchain-Based AC | [70], [71], [72], [73] |
| **Encryption** | IBE | [79], [90],[91] |
| | ABE | [75], [76], [77], [78], [86], [87], [92] |
| | RBE | [80], [81] |
| **Network Segmentation** | Micro-segmentation | [93], [7], [94], [95], [96], [97], [98] |
| | SDP | [99], [100], [101], [102] |
| **Automation and Orchestration** | - | [103], [104], [105], [106] |
| **Monitoring and Analytics** | Threat Intelligence | [107],[108], [109], [110], [111], [112], [113], [114], [115] |
| | Anomaly Detection | [116], [117], [125], [118], [119], [120], [121], [122], [123], [124], [125] |

review literature focuses on monitoring while only some of them focus on analytics or both.

Section 3.1 has provided a comprehensive review of current research on techniques to implement ZTA tenets. The overview of the techniques, their granularity, and the reviewed references can be found in Table 3.6.

## 3.2 Evaluation of ZTA

Compared to traditional security models, ZTA represents a significant shift in security configuration for cloud environments. However, it is also markedly more complex than traditional security models, requiring a holistic approach to security management. This section evaluates ZTA from various perspectives, starting with its effectiveness, followed by ease of use, and finally, implementation overhead and complexities.

1. **Does ZTA eliminate a greater number of threats, improving the overall threat landscape, or is it similar to other security models?**

   ZTA presents a paradigm shift in cybersecurity by adopting a "never trust, always verify" approach, fundamentally challenging traditional perimeter-based security models. By scrutinizing every access request and continuously verifying the identity and trustworthiness of users and devices, ZTA significantly reduces the attack surface, thus mitigating a broader range of threats. This approach contrasts with traditional models that rely heavily on perimeter defenses, often proving inadequate against insider threats or sophisticated attacks. Consequently, ZTA demonstrates a clear superiority in threat mitigation, improving the overall security landscape by proactively minimizing vulnerabilities and enhancing resilience against evolving cyber threats.

   However, while ZTA excels in addressing modern cybersecurity challenges, it's not entirely foolproof. As with any security framework, ZTA requires careful implementation and ongoing management to remain effective. Moreover, some critics argue that ZTA may not eliminate all threats, especially those related to human error or social engineering attacks. For instance, even with robust ZTA measures, phishing attacks can still trick users into divulging sensitive information, bypassing technical defenses. Additionally, misconfigurations or lapses in policy enforcement can create vulnerabilities that attackers might exploit. Therefore, organizations must adopt a comprehensive security strategy that includes technological and human-centric approaches [38].

2. **Does ZTA improve the user experience or make it cumbersome for users to access resources?**

   ZTA aims to balance robust security measures with optimized user experience, enhancing security without unduly burdening users. Although ZTA introduces additional authentication and authorization steps, it ultimately strengthens security. However, the initial setup and adoption of ZTA principles may introduce complexities and friction, such as additional authentication steps or device authorization approvals, potentially disrupting workflows. Organizations must invest in user training and support to ensure a smooth transition and ongoing usability [126].

A major concern for organizations adopting ZTA is its impact on operations and end-user experience. ZTA's additional authentication and access controls could lead to repeated user authentications based on resource access and enterprise security policies. While ZTA should ideally enhance security transparently, careful design and testing are essential to minimize disruptions and ensure a seamless user experience. A common understanding across the organization regarding ZTA, its benefits, and gauging ZTA maturity is vital [127, 128]. Determining the most suitable ZTA approach requires thoroughly assessing the organization's security posture, risk tolerance, and business objectives. Successful adoption and implementation necessitate collaboration among IT, security, and business teams, ensuring a shared understanding of ZTA and its implications.

3. **Does ZTA add operational overhead, including deployment, management complexities, and hidden costs?**

   While ZTA offers significant security advantages, its implementation may entail operational overhead, including management complexities and potential hidden costs. Continuous monitoring and updating access policies to align with organizational changes and evolving threats also contribute to operational overhead as the organization grows. Moreover, training staff on ZTA principles and tools, alongside infrastructure setup and maintenance costs, should be factored into the overall operational considerations. Despite these challenges, the enhanced security posture and streamlined access offered by ZTA justify the investment for many organizations, provided proper planning and resource allocation are undertaken [129].

   ZTA involves pervasive, fine-grained, and continuous security controls like authentication and access control, making security management more complex. It requires extensive attribute-based policies that must be trustworthy, correct, consistent, minimal, and complete. Challenges include formalizing and automating risk-based criteria, ensuring policy correctness and consistency, eliminating redundant policies, and managing policy completeness. Additionally, ZTA impacts application performance, complicates legacy system upgrades, and integrates with cloud infrastructures. Addressing these issues necessitates technical solutions and incremental deployments, potentially resulting in isolated ZTA segments within systems [130].

## 3.3 Challenges Associated with ZTA

A qualitative evaluation of ZTA, as presented in the previous section, highlights several potential challenges with real-world implementation. This section outlines the

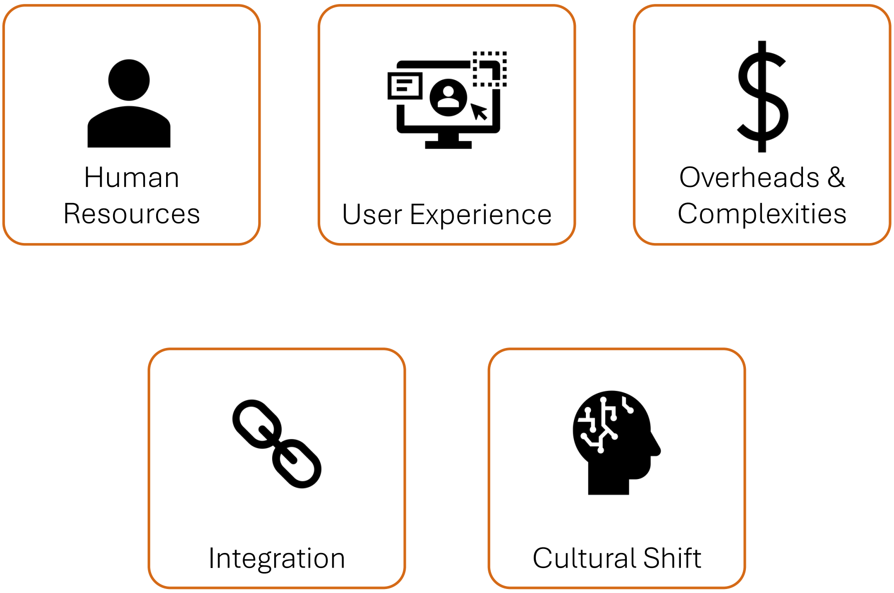



**Figure 3.2:** Challenges with ZTA

technical, operational, and organizational obstacles that organizations must overcome to ensure a successful implementation and transition to ZTA.

### 3.3.1 Human Resources

The shift to ZTA necessitates a workforce skilled in new technologies and approaches. Implementing ZTA requires specialized knowledge in advanced cybersecurity protocols, micro-segmentation, Secure Access Service Edge (SASE) frameworks, network security, identity and access management, and machine learning. Finding and retaining personnel with these skills can be challenging and costly. IT staff may require extensive training to understand and manage ZTA. This can include formal education, certifications, and hands-on training sessions. ZTA implementation often involves integrating various IT disciplines, necessitating a broad skill set that includes network engineering, cybersecurity, software development, and systems administration.

The allocation of technological and human security resources varies depending on the industry and size of the organization. Larger firms, particularly those in critical sectors such as finance and telecommunications, typically invest heavily in human security resources, often employing a Chief Information Security Officer (CISO) and maintaining dedicated information security departments. This emphasis on human security resources is increasingly prevalent across industries and among smaller firms, reflecting a growing recognition of the importance of safeguarding information. However, while there's a clear distinction between human non-security IT resources and human security IT resources, the line between technological non-security IT resources and technological security IT resources can be less defined in practice. This lack of differentiation presents economic challenges, as it becomes difficult to delineate between general IT and IT security budgets accurately. This blurring of lines underscores the need for clearer delineation and allocation of resources to address security concerns while managing costs effectively [126].

### 3.3.2 User Experience

The impact of ZTA implementation on user experience is a critical consideration for organizations. While ZTA offers enhanced security through continuous authentication, this reliance on frequent login prompts and verification steps can create friction in the user experience. Strict access controls and verification processes can also delay resource access, impacting productivity and hindering efficiency.

Moreover, users' resistance to the new security measures can pose significant challenges for ZTA adoption. Perceived inconveniences such as additional authentication steps or restricted access to resources may lead to resistance and non-compliance with security protocols. This resistance undermines the effectiveness of ZTA and creates security vulnerabilities as users seek workarounds to bypass security measures. Therefore, organizations must carefully balance security requirements with user experience considerations to ensure that ZTA implementation is effective and user-friendly, fostering a culture of security awareness and cooperation among users.

### 3.3.3 Overhead and Complexities

Implementing ZTA introduces overhead costs and complexities, including infrastructure upgrades, security tools and technologies, ongoing maintenance and management, and compliance and audit. Substantial investments may be needed for new hardware, software, and network components compatible with zero trust protocols and advanced security tools like IAM systems, MFA, and continuous monitoring solutions. Maintaining a zero trust environment requires management of a complex environment due to constant updates, threat intelligence integration, and policy adjustments. Ensuring that ZTA meets regulatory compliance standards can involve significant expenditures on audits and certifications. However, research shows that ZTA implementation can significantly reduce risk impact over a span of four years for various organizations [129, 126]. Although the initial cost might be high for implementing ZTA, it will be offset by the long-term benefits of reduced risk exposure and potential financial losses.

### 3.3.4 Integration

Integrating ZTA into an existing IT ecosystem presents substantial challenges, particularly with legacy systems. Furthermore, vendor lock-in presents a significant challenge during ZTA implementation.

#### Legacy Systems

Integrating ZTA with legacy systems presents significant challenges, as these older systems are often not designed to support modern security technologies such as micro-segmentation and network traffic analysis. Additionally, legacy systems may run outdated software and hardware, increasing their cyber-attack vulnerability. For instance, relying solely on password-based authentication systems is inadequate, and incorporating some form of multi-factor authentication is essential to enhance security [131], but not all legacy systems support MFA. This can necessitate significant modifications or even the complete replacement of outdated systems, resulting in increased costs and complexity. Moreover, updating or replacing legacy systems to support ZTA introduces the risk of system downtime, which can disrupt business operations.

Organizations must carefully balance the need for enhanced security with budget constraints and the practicalities of minimizing disruptions to critical processes during integration. Additionally, data migration from legacy systems to new, secure environments poses risks such as data loss, corruption, and potential breaches, highlighting the importance of meticulous planning and risk mitigation strategies.

#### Vendor Lock-In

Organizations often rely on specific vendors for security solutions and technologies, leading to dependency on these vendors for updates, support, and compliance. This dependency can restrict flexibility and hinder the organization's ability to adapt to evolving security needs. Interoperability challenges further complicate the integration process, as different vendors may utilize proprietary technologies that do not easily integrate with each other. The study [132] indicates that many customers, including decision-makers in companies, lack awareness about proprietary technologies that could hinder interoperability and portability when acquiring cloud services from providers. This can result in fragmented security measures and potential security gaps, undermining the effectiveness of ZTA. Moreover, vendor lock-in can lead to higher costs over time, as organizations may face increased charges for support, updates, and additional features, highlighting the importance of carefully evaluating vendor relationships and considering long-term implications during the ZTA implementation process.

### 3.3.5 Cultural Shift

Adopting ZTA requires a significant cultural transformation within an organization. Transitioning to a zero trust model entails challenging long-standing security practices and ingrained mindsets. This requires effective change management strategies to ensure smooth adoption and buy-in from all stakeholders. Central to this process is educating employees about the importance of zero-trust principles and providing guidance on navigating the new security framework. Continuous education and awareness initiatives are pivotal in ensuring that all organization members understand their role in upholding security standards. The study [133] suggests that the training program boosts the motivation of employees to learn irrespective of their background in security. Employees and management may resist the cultural change due to comfort with existing practices and fear of the unknown. Overcoming this resistance requires clear communication, demonstrating the benefits of ZTA, and involving employees in the transition process.

This chapter has presented a comprehensive review of ZTA and the associated challenges. The next chapter will focus on how to apply the discussed techniques to secure AWS environments.

# Chapter 4
# AWS Security through ZTA

AWS offers a comprehensive suite of services that enable organizations to leverage cloud computing for a variety of applications and workloads. However, the inherent nature of cloud computing introduces significant security challenges and risks that organizations must address to protect their assets and comply with regulations. These challenges range from unauthorized access and data breaches to misconfigurations and insider threats, making the cloud security landscape complex and ever-evolving. As businesses migrate to the AWS, enhancing security measures to protect sensitive data and applications becomes increasingly critical.

ZTA provides a foundational approach to enhance cloud security by adopting its principles. The principles, such as continuous authentication and least privilege access, ensure that every access attempt is verified and that users only have access to necessary resources. ZTA also promotes the segmentation of networks into smaller, isolated zones, limiting the impact of potential breaches.

AWS offers tools that can help organizations implement various ZTA principles. These tools include AWS Key Management Service (KMS), AWS Nitro System, and API gateways, which provide essential tools to protect cloud environments. There are also solutions provided by many third-party vendors to implement ZTA principles in AWS environments. ZTA principles can be systematically integrated with security tools provided by AWS and third-party vendors, creating a robust security framework that effectively mitigates risks and ensures the protection of their cloud environments.

This chapter first describes various aspects of AWS security. Then, it explains how various ZTA principles can effectively address the corresponding security issues for each aspect. Moreover, it explores tools provided by AWS to implement various principles of ZTA. Furthermore, since ZTA implementation from the ground up can be challenging and may require significant effort, this section explores various off-the-shelf ZTA solutions provided by various third-party vendors towards the end.

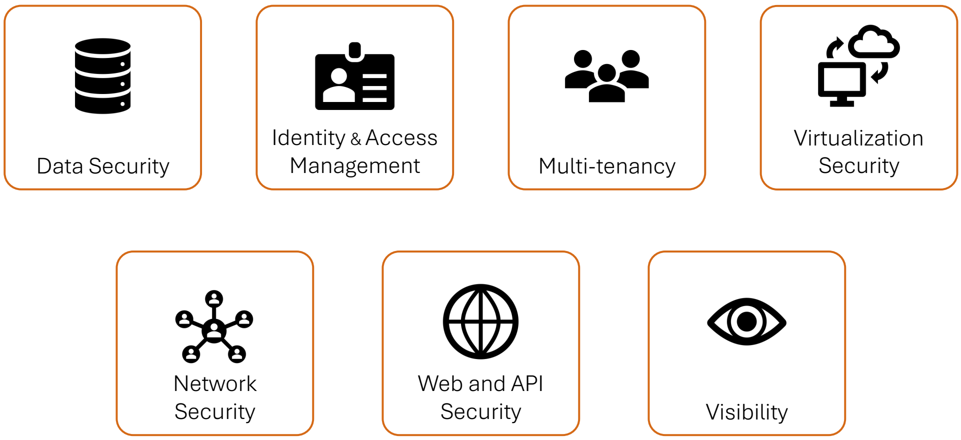



**Figure 4.1:** Aspects of AWS Security

## 4.1 Various Aspects of AWS Security and Exploration of ZTA

Maintaining robust security is essential for protecting data integrity, securing resources, and ensuring operational continuity within AWS environments. The critical security aspects within AWS environments are illustrated in Figure 4.1. Each security aspect presents distinct challenges that impact the overall security posture within the AWS environment. ZTA offers techniques to strategically implement its tenets to mitigate these challenges effectively.

This section provides a detailed examination of various security facets within AWS environments, highlighting specific challenges associated with each aspect. It explores how ZTA methodologies can be leveraged to address these challenges, ensuring robust security measures to safeguard AWS environments.

### 4.1.1 Data Security

Data security in AWS is crucial given the criticality of data as a company's most valuable asset and the increasing migration of data to AWS environments. The lifecycle of data in the AWS encompasses various stages such as production, storage, processing, and deletion, each requiring robust security measures to ensure availability, integrity, and confidentiality. Securing data in AWS involves addressing challenges like data breaches, data loss, unauthorized access, and compliance with data protection regulations.

Data breaches pose significant risks in AWS environments, where unauthorized access to sensitive data can lead to financial losses, reputational damage, and legal repercussions. Breaches often exploit vulnerabilities in infrastructure, application design flaws, insufficient authorization controls, and inadequate audit mechanisms. On the other hand, data loss can occur due to accidental deletion, malware attacks, encryption key loss, system failures, or natural disasters, jeopardizing data integrity and operational continuity [134].

Data segregation risks arise from inadequately secure boundaries, poorly designed virtual machines, and client-side hypervisors. This issue is particularly challenging in the AWS environment, where shared resources can compromise data dependability if not properly managed [135]. Malware and injection attacks on AWS applications can also result in data destruction. An Advanced Persistent Threat (APT) is another type of a prolonged and targeted cyberattack where an unauthorized user gains access to a system and remains undetected for an extended period, posing significant enterprise risks such as data theft [136].

The shared environment of AWS platform raises concerns about data privacy, and it can cause uncertainty regarding the precise physical location of the data, potentially conflicting with data protection laws and legal jurisdictions. Storing data in AWS services like S3 buckets, Elastic Block Store (EBS) volumes, and RDS instances without proper encryption poses a major security risk. If these services are compromised, unauthorized parties can easily access unencrypted data, resulting in breaches and compliance violations. Amazon's S3 does not encrypt data by default; therefore, users must configure mechanisms to encrypt their data. Malicious users can exploit weaknesses in the data security model to gain unauthorized access and compromise data integrity.

ZTA techniques such as authentication, access control, encryption, and monitoring and analytics are instrumental in mitigating these security challenges. Authentication can ensure that only authorized users and devices can access data, verifying identities continuously rather than trusting any entity by default. Access control can limit permissions strictly based on the principle of least privilege, reducing the attack surface and preventing unauthorized data access. Encryption can protect data at rest and in transit, safeguarding against unauthorized viewing or tampering. It helps with secure data backup and recovery protocols. Monitoring and analytics can help detect anomalies in database transactions.

Implementing ZTA techniques in AWS enhances data security by establishing a framework where every access attempt is rigorously authenticated, access permissions are finely controlled, and data is encrypted to prevent interception or unauthorized access. By adhering to these principles, organizations can bolster their data security posture in AWS, mitigating risks associated with data breaches and ensuring compliance with regulatory requirements.

### 4.1.2 IAM

IAM is a critical framework in AWS that encompasses organizational policies for managing digital identities, authenticating users, devices, or services, and granting or denying access rights to data and system resources. IAM systems must incorporate comprehensive controls to save and record user login information, manage an enterprise database of user identities, and handle the assignment and revocation of access privileges. Essentially, an IAM system establishes a centralized directory service that addresses all aspects of the company's user base.

Managing appropriate access control mechanisms to deter unauthorized individuals from accessing resources and data presents a significant challenge within AWS environments characterized by intricate access demands. Security breaches can occur due to misconfigured IAM policies, excessive permissions granted to users or roles, weak authentication mechanisms, or compromised credentials [136]. In an AWS environment, basic authentication mechanisms are not enough for accessing resources.

Privileged Access Management (PAM) is a specialized branch of IAM that concentrates on securing privileged accounts and systems. As an information security mechanism, PAM protects identities with elevated access or capabilities that exceed those of regular users. However, privileged user access comes with inherent security risks, such as potential misuse or abuse of elevated permissions, insider threats, and increased susceptibility to cyberattacks. Therefore, it is crucial to ensure the safe management of privileged accounts.

In contemporary settings, many small-scale companies use Lightweight Directory Access Protocol (LDAP) servers for tasks related to authorization and authentication. However, credentials are frequently stored on the cloud provider's premises rather than within the organization to take advantage of centralized management, scalability, and enhanced accessibility. Despite these benefits, this practice can lead to significant issues; if the cloud service becomes inaccessible, users may be unable to maintain connectivity and authorization to any applications, disrupting business operations [135].

ZTA techniques such as authentication, access control, and automation and orchestration can significantly enhance IAM in AWS. MFA and continuous authentication with correctly configured access control ensure that only authorized users can access resources. This involves continuously verifying identities and using MFA rather than blindly trusting any entity by default. The access control can strictly limit permissions based on the principle of least privilege, reducing the risk of unauthorized access. Automation and orchestration can streamline the management of IAM policies, ensuring that security measures are consistently applied and updated across the entire AWS environment. Organizations can bolster their IAM strategies by integrating these ZTA techniques, effectively mitigating security challenges and ensuring a more secure AWS environment.

### 4.1.3 Multi-Tenancy

Multi-tenancy is a fundamental aspect of cloud computing. It allows multiple users or tenants to share the same cloud infrastructure while maintaining separate environments. It can occur at various levels: between different companies using the same cloud provider and potentially sharing the same or connected hardware; between different teams within a single organization sharing the same or connected cloud hardware; and between different projects within the same team, with one cloud resource shared across multiple projects. Each scenario introduces data exposure, interference, and resource contention challenges.

Multi-tenancy in cloud environments presents significant advantages and challenges. It introduces potential issues such as data exposure, interference, and resource contention. The risk of multi-tenancy lies in the co-residency of data and resources from different clients on shared physical hardware, resulting in the possibility of data exposure. This compromises the privacy and confidentiality of sensitive information stored within cloud environments [137].

In AWS, multi-tenancy enables different users to utilize a single platform simultaneously, providing significant cost savings and resource efficiency. For example, in an IaaS service, Virtual Machine Monitors (VMMs) facilitate the sharing of resources, with each tenant running their instances as VMs. In a PaaS service, a virtual platform supports multiple applications within a multi-tenant environment, such as Java Virtual Machines and .NET frameworks.

Multi-tenancy can make the applications vulnerable to various security threats, such as co-tenancy, co-location, and co-residence attacks, due to the proximity of valuable customer data in the same physical location. These attacks allow attackers to access neighboring VMs or running applications. Additionally, multi-tenancy environments are susceptible to DoS attacks, where an attacker can consume excessive resources, disrupting service availability for other tenants [138].

IaaS services in a multi-tenant environment using virtualization enable resource sharing among multiple users. However, in such environments, the hypervisor can potentially allow a malicious user to access information about other users. This poses a significant threat since the infrastructure is often not designed to provide strong isolation in a multi-tenant setting. Sharing resources can compromise the overall cloud infrastructure by exposing one user's information to another [138].

The multi-tenant nature also introduces resource contention problems when some tenants use hardware disproportionately, whether due to demand or malicious activities. Moreover, interference can arise when one tenant's actions negatively affect another tenant's environment. Additionally, change management becomes riskier; configuration changes for one tenant may inadvertently impact others [139].

ZTA principles offer solutions to these multi-tenancy challenges. Implementing strict access control ensures that each tenant can only access their data and resources, reducing unauthorized access risks. Encryption safeguards data at rest and in transit, ensuring that exposed data remains unreadable to unauthorized parties. Network segmentation isolates tenant environments, mitigating the risk of co-residency attacks and interference from neighboring tenants. By integrating these ZTA techniques, organizations can enhance the security and integrity of their multi-tenant virtual AWS environments, providing a secure and reliable cloud experience for all users. However, ZTA does not address infrastructure-level isolation since it's managed by AWS.

### 4.1.4 Virtualization Security

The widespread adoption of cloud computing in various industries can be attributed to its virtualized infrastructure. AWS relies heavily on VMs such as EC2 and virtualization technology to develop services for business purposes. While virtualization offers significant benefits, it also introduces vulnerabilities to potential threats and attacks.

Virtualization software, used to create virtualized services and images, must contend with various viruses that can potentially compromise or disrupt the integrity of virtualized code, and security vulnerabilities in this software can lead to data leaks, allowing malicious individuals to bypass security restrictions and gain unauthorized privileges. Ongoing research aims to enhance logical and virtual isolation to mitigate these risks [138].

Virtualization environments are also susceptible to DoS attacks, where a guest machine consumes excessive resources, disrupting other guests and the host. Virtual Machine (VM) sprawl, caused by poor VM management policies, leads to an uncontrolled growth of VMs, many of which remain idle, wasting the host's resources. These vulnerabilities highlight the importance of robust security measures in virtualization, including strong isolation, resource management, and protection of the host machine to ensure a secure virtual environment [140].

Vulnerabilities targeting the Operating System (OS) kernel of the host where the malicious container operates are particularly concerning. Through maliciously crafted syscalls, adversaries can initiate actions such as triggering memory leaks, writing arbitrary data to shared files with the host, or gaining elevated privileges within the host. One significant example involves vulnerabilities in system calls, which facilitated privilege escalation attacks granting access to the host. The primary issue contributing to these vulnerabilities is the excessive default privileges granted to containers. While Linux kernel security modules have been developed to confine container capabilities, they do not effectively address the challenge of minimizing container privileges. Addressing this requires leveraging tools capable of automatically identifying the minimal privileges necessary for containers to execute their applications correctly while reducing their dependency on interactions with the OS kernel [140].

Improper isolation between VMs can lead to data leakage or unauthorized interactions. Ensuring secure management of virtualized resources and protecting against virtual machine escape attacks are critical concerns [141]. While applications run on virtualized servers that appear isolated to users, the servers themselves must also be securely isolated from each other. Prior research [142] has demonstrated that adversaries who gain access to a container within a host belonging to a large cluster can exploit vulnerabilities across different layers to execute potent attacks.

AWS hosts client applications using VMs within a shared infrastructure enabled by virtualization technology. These VMs operate on the physical infrastructure of the AWS. Hypervisors manage these VMs, handling virtual memory and CPU scheduling policies. Since hypervisors are crucial for managing policies in virtualization, they are prime targets for intruders seeking control. Attackers often attempt co-location attacks to compromise services in AWS environments. Such attacks can damage VMs and the underlying infrastructure. Hypervisor vulnerabilities can also be exploited to gain unauthorized access to VMs.

ZTA principles effectively solve these virtualization security challenges. Implementing access control ensures that only authorized users can interact with VMs and the hypervisor, minimizing the risk of unauthorized access. Encryption protects data within VMs and during transfer between VMs and the host, reducing the risk of data breaches. Network Segmentation isolates different VMs and network segments, preventing the spread of attacks and ensuring that a compromised VM does not jeopardize the entire infrastructure. By leveraging these ZTA tenets, organizations can enhance the security of their virtualized AWS environments, mitigating risks and protecting their assets.

### 4.1.5 Network Security

Interaction with AWS relies extensively on internet connectivity and servers that store data for diverse applications, underscoring the critical role of network security. This security domain is pivotal for ensuring the functionality and connectivity of AWS applications, with responsibilities distributed between AWS and users. However, organizations might inadvertently overlook certain security measures as cloud operations scale, exposing vulnerabilities that malicious entities can exploit. These oversights contribute to a heightened risk of network threats, compromising AWS services' overall integrity and reliability [134].

AWS services face significant vulnerability to Distributed Denial-of-Service (DDoS) attacks [143], which inundate systems with excessive traffic, leading to service disruptions and potential reputational harm. These attacks, exemplified by cloud zombie attacks involving numerous compromised devices flooding networks, can overwhelm AWS infrastructure, disrupting services reliant on uninterrupted network connectivity and availability [144]. Advanced DDoS variants like the Yo-Yo attack exploit auto-scaling features, causing economic losses through repeated oscillations between scale-out and scale-in states [145].

Another variant of DDoS an attack is the Tandem attack, where an attacker can exploit the tandem behavior of microservices, particularly through different auto-scaling mechanisms. Unlike large-volume DDoS attacks that target network infrastructure, the Tandem Attack focuses on the application layers, requiring significantly fewer resources from the attacker [146]. Link Flooding Attacks (LFAs) further disrupt network connectivity by congesting critical links with a high volume of flows, complicating mitigation efforts for network operators. Adversaries exploit these attacks by blending low-volume, legitimate flows with malicious traffic, challenging the implementation of effective countermeasures [147].

The decentralized architecture of AWS services increases susceptibility to Man-in-the-Middle (MitM) attacks and exposes them to significant malware threats. MitM attacks occur when malicious actors intercept and alter communication between parties undetected, exploiting weaknesses in communication channels, compromised certificates, or inadequate encryption protocols [148]. These vulnerabilities can lead to data leakage, unauthorized access, or manipulation of sensitive information within AWS environments. Concurrently, the decentralized structure of AWS services enables malware to infiltrate specific points in the infrastructure and propagate through interconnected components. Exploiting inherent vulnerabilities in AWS architecture, such malware poses substantial risks to AWS systems' overall security and stability [149].

Further vulnerabilities persist in AWS environments due to low adoption of

HyperText Transfer Protocol Secure (HTTPS). When HyperText Transfer Protocol (HTTP) is used instead, the lack of encryption leaves cloud services vulnerable to MitM attacks, where data can be intercepted and altered. Additionally, the exposure of hashed passwords in some protocols without HTTPS can facilitate Rainbow Table attacks, increasing the risks of data breaches and unauthorized access. However, using HTTPS can mitigate risks by ensuring secure data transmission and preventing MitM attacks. [135].

Brute force and dictionary attacks target weak passwords, making systems vulnerable to unauthorized access. Additionally, DNS attacks pose significant threats by redirecting traffic to malicious sites. To enhance security, implementing DNS security measures such as Domain Name System Security Extensions (DNSSEC) and binding DNS records to HTTPS certificates can provide additional layers of protection.

In SaaS, organizational data resides outside traditional security boundaries, which might expose sensitive information. Therefore, in SaaS environments, securing data flow through robust encryption techniques such as Secure Sockets Layer (SSL) and TLS is essential to protect against unauthorized access and ensure data integrity [135]. AWS addresses these issues by enforcing SSL endpoints for secure data transmission, although vulnerabilities such as packet sniffing persist. Furthermore, SaaS models must implement robust encryption techniques and comprehensive security protocols to safeguard data integrity, underscoring the critical role of administrators in establishing and auditing security measures for AWS environments.

Within AWS, protecting network infrastructure and communication channels is paramount to prevent data interception and DDoS attacks. Common concerns include inadequate encryption protocols, weak network segmentation, and misconfigurations in network devices or protocols, which highlight the ongoing challenges in maintaining network security [150, 151, 152].

In AWS environments, ZTA offers critical advantages by implementing its tenets through encryption, network segmentation, and monitoring and analytics. Encryption can ensure data integrity and confidentiality during transmission, protecting against unauthorized access. Network segmentation can limit the impact of breaches by isolating network segments and reducing the attack surface. Monitoring and Analytics provide continuous oversight, enabling proactive threat detection and rapid response to potential security incidents. By implementing ZTA principles, AWS environments can strengthen their security posture, mitigate risks, and maintain the reliability of cloud services.

### 4.1.6 Web Application and API Security

Web applications within AWS serve as crucial components for delivering dynamic and interactive services to users. These applications typically run on different platforms and frameworks, each with its own set of unique vulnerabilities. In the context of cloud computing, the complexity increases as these applications may involve numerous software modules written in various programming languages by multiple developers, further introducing potential security risks.

APIs in AWS are critical interfaces enabling communication between software applications and cloud services. APIs serve as gateways for accessing network functionalities, provisioning, monitoring, and managing cloud services. Given their pivotal role, the security of APIs is paramount to protect against unauthorized access, data breaches, and other cyber threats. Insecure APIs can arise from broken authentication, unrestricted resource access, and security misconfigurations, making them attractive targets for malicious actors.

Securing web applications and APIs involves defending against a range of threats such as injection attacks (e.g., Structured Query Language (SQL) injection, Cross-Site Scripting (XSS)), authentication and session management vulnerabilities, and insecure API endpoints [152]. SQL injection attacks specifically exploit weaknesses in back-end applications, targeting the interaction between the application and its database. Trust mechanisms in web-based business services are crucial for protecting sensitive information, although these services also present opportunities for malicious attacks. OSs play a critical role in security, with vulnerabilities across desktop, server, network, and smartphone OSs introducing multiple security challenges.

Authentication and session management vulnerabilities can lead to unauthorized access and session hijacking. Service and account hijacking involves unauthorized access to AWS accounts through methods such as phishing. This compromises data confidentiality and integrity. Social engineering attacks deceive users into revealing personal information, leading to unauthorized account access. Insider threats involve malicious actions by employees or other insiders familiar with the organization's security protocols. These malicious insiders misuse their authorized access to leak or manipulate data, posing significant internal security risks. Other notable threats include crypto-jacking, which exploits computers to mine cryptocurrency without the owner's consent.

Bearer tokens are widely used for simplicity in API security, but they come with inherent risks if not handled properly [153]. They are susceptible to interception and abuse if leaked or stolen, compromising the security of API endpoints. One way to significantly enhance API security is through the use of asymmetric cryptography, where clients use their private keys to sign API requests. This method ensures that requests originate from authorized clients and have not been tampered with in transit. By verifying the digital signature on each request, API servers can authenticate and validate the integrity of incoming data, thereby reducing the risk of unauthorized access or data manipulation.

In API security, HTTPS plays a crucial role in ensuring the confidentiality, integrity, and authenticity of data transmitted between clients and servers. APIs often handle sensitive information, such as user credentials or personal data, making them prime targets for attackers. Utilizing HTTPS for API communications encrypts data in transit, preventing eavesdropping and tampering by malicious actors. However, the security provided by HTTPS hinges on configuring its cryptographic algorithms, protocol versions, and cipher suites. Outdated or weak configurations can expose APIs to vulnerabilities like MitM attacks or decryption attempts.

API security is essential for controlling access to AWS services. APIs enable developers to monitor, provision, and manage resources, but weak APIs can lead to serious issues such as unauthorized configuration changes, data leaks, and server disablement. Access to AWS service APIs should be restricted using encrypted keys for user authentication, which should be securely stored in a hardware security module [139].

Implementing a ZTA in AWS environments can significantly enhance the security of web applications and APIs. ZTA techniques, such as encryption, network segmentation, and monitoring and analytics, can mitigate various security challenges. Encryption can ensure that data in transit and at rest is protected against unauthorized access. Network segmentation can limit the potential impact of breaches by isolating different segments of the network, preventing lateral movement by attackers. Continuous monitoring and analytics enable the detection of suspicious activities and potential threats in real time, allowing for swift response and mitigation. By adopting ZTA principles, organizations can create a more secure and resilient cloud environment, protecting their web applications and APIs from evolving cyber threats.

### 4.1.7 Visibility

One of the other challenges in cloud security is the limited visibility into cloud operations. Cloud service providers abstract much of the operations from users, making it difficult for organizations to fully understand and monitor the security posture of their cloud environments. This lack of visibility can lead to gaps in threat detection, making it challenging to promptly identify and respond to security incidents. Additionally, it hinders compliance efforts as organizations struggle to demonstrate adherence to regulatory requirements without comprehensive visibility into the underlying deployments [154].

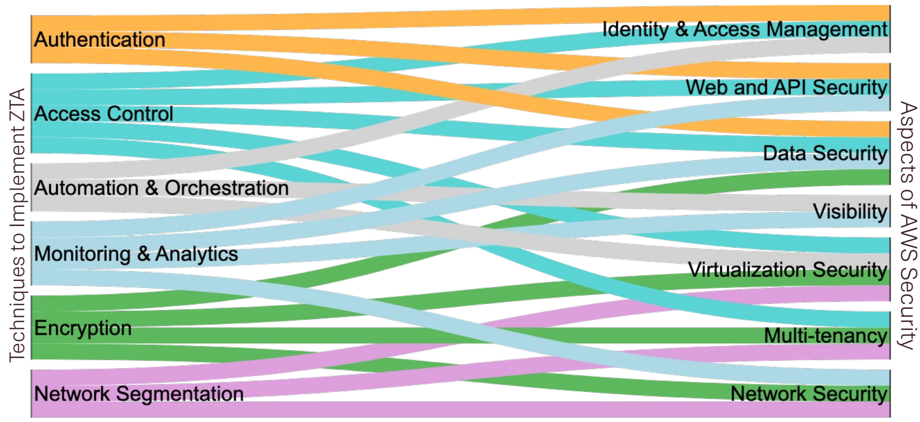



**Figure 4.2:** Techniques to Implement ZTA: Aspects of AWS Security

Businesses today have limited control because the service provider is responsible for managing the entire infrastructure, including security, without providing full transparency. This lack of control extends to critical aspects such as data location, access policies, and security protocols, often managed exclusively by the cloud provider. Organizations must transition significantly as they delegate IT control to a third party when using cloud services. This delegation necessitates a shift in trust and reliance on the cloud provider's security measures, which can be a source of concern for many businesses. Moreover, the dynamic nature of cloud environments, with frequent updates and changes, further complicates the ability of organizations to maintain a consistent and robust security posture [135].

Implementing ZTA tenets such as automation and orchestration and monitoring and analytics can significantly mitigate these visibility challenges. Automation and orchestration streamline security processes, ensuring consistent application of security policies and enabling rapid response to potential threats. By automating repetitive tasks, organizations can maintain a high level of security without manual intervention, reducing the likelihood of human error. Continuous monitoring and analytics provide real-time insights into the cloud environment, enabling organizations to promptly detect anomalies and potential security incidents. These capabilities enhance situational awareness and facilitate compliance by providing detailed visibility into security practices and operations. However, ZTA does not provide visibility into or control over the physical location of the data.

This section explored the security challenges in AWS environments and how ZTA can be utilized to address these challenges. Figure 4.2 provides an overview mapping between techniques to implement ZTA and the various security aspects of AWS.

## 4.2 AWS Tools to Realize ZTA Principles

To address various security challenges, AWS offers a variety of tools to its customers that they can integrate into their cloud applications. Each tool can help realize the principles of ZTA and enhance overall security. Table 4.1 lists various aspects of AWS security, specific security challenges associated with these aspects, and the tools offered by AWS to address these challenges. This section discusses the tools provided by AWS to implement ZTA.

AWS provides robust SSE options to secure data at rest across various services, including S3, EBS, and RDS. For S3, AWS offers multiple encryption methods: SSE-S3, where Amazon S3 manages the keys; SSE-KMS, which uses customer-managed keys in AWS KMS; and SSE-C, which allows customers to provide their own keys. RDS supports encryption at rest with both AWS-managed and customer-managed keys, encryption in transit using SSL/TLS, and Transparent Data Encryption (TDE) for Oracle and SQL Server. EBS encryption can also be enabled with AWS-managed or customer-managed keys while creating new volumes or snapshots. Additionally, AWS DynamoDB automatically encrypts all data at rest, ensuring comprehensive data protection across services.

AWS IAM allows fine-grained access control through detailed policies and roles, enabling administrators to grant permissions as needed. For enhanced security, AWS offers MFA through AWS Cognito. Additionally, AWS Client VPN provides mutual authentication between clients and servers, ensuring secure connections. AWS KMS handles key management and encryption, while AWS CloudHSM offers hardware security modules to secure encryption keys and provide comprehensive key management solutions.

AWS employs a custom, high-security hypervisor for EC2 instances called the AWS Nitro System, which isolates and protects each instance by integrating hardware and software security measures. Nitro Enclave ensures strong tenant isolation through advanced virtualization and network controls. Services like Amazon EC2 utilize instance isolation techniques, preventing one tenant from accessing another's data and enhancing overall security in a multi-tenant environment. Additionally, Nitro Enclaves uses cryptographic attestation to verify the integrity of the enclave environment, ensuring only authorized code runs within the enclave. Integration with AWS KMS allows for secure key management and data encryption.

AWS provides a suite of tools to enhance network security, including VPC for

**Table 4.1:** AWS Tools to Realize ZTA Principles

| **Security Categories** | **Affected by** | **AWS Solutions** |
|---|---|---|
| Data Security | – Data breaches<br>– Unauthorized access<br>– Data loss<br>– Account hijacking<br>– Data locality | AWS KMS, Amazon S3 Server-Side Encryption (SSE), AWS DynamoDB, AWS CloudHSM |
| Identity and Access Management | – Unauthorized access<br>– Privilege escalation | AWS IAM, AWS Cognito, AWS Client VPN |
| Multi-tenancy | – Information leakage between tenants<br>– Resource contention | Amazon EC2, AWS Nitro System |
| Virtualization Security | – Hypervisor vulnerabilities<br>– VM escape attacks<br>– VM sprawls<br>– DoS attacks | Amazon EC2, AWS Nitro System |
| Network Security | – Network attacks e.g DDoS attacks<br>– Unauthorized network access<br>– Session Hijacking | AWS Virtual Private Cloud (VPC), Security Groups and Network Access Control Lists (ACLs), AWS WAF, AWS Shield |
| Web Application and API Security | – API vulnerabilities<br>– Injection attacks<br>– Insecure coding practices | AWS API Gateway, AWS Lambda, AWS App Mesh |
| Visibility | – Lack of monitoring and logging<br>– Insufficient visibility into resource usage | AWS CloudWatch, AWS CloudTrail, AWS Config, AWS GuardDuty |

network isolation, Security Groups, and Network ACLs for traffic control, AWS WAF for application-level protection, and AWS Shield for DDoS protection.

The AWS API Gateway facilitates secure and scalable API management, enabling organizations to create, publish, maintain, monitor, and secure APIs at any scale. This helps implement fine-grained access control and monitor traffic between services. AWS Lambda provides serverless computing, allowing code execution in response to events without provisioning or managing servers. This reduces the attack surface and simplifies the implementation of microservices architectures. AWS App Mesh provides a service mesh that ensures secure and reliable communication between microservices. It enhances observability and control, enabling seamless encryption of service-to-service communication, and provides rich telemetry to monitor and debug applications.

For monitoring and visibility, AWS offers CloudWatch and CloudTrail, which provide comprehensive monitoring, logging, and visibility into AWS environment activities and resource usage. AWS Config allows continuous monitoring of resource configurations and compliance, while GuardDuty offers automated threat detection. These tools enable detailed logging, auditing of API calls, and compliance monitoring to ensure a secure and compliant cloud environment.

## 4.3 ZTA Solutions Offered by Different Vendors for AWS

Implementing ZTA in practice can be challenging due to the complexity of restructuring existing network architectures and implementing new security protocols. Organizations must meticulously assess and redefine access policies, integrate MFA systems, and establish continuous monitoring mechanisms. Additionally, implementing micro-segmentation requires careful planning to define and enforce granular access controls across different network segments. To navigate these challenges, various service providers offer pre-built ZTA solutions that streamline implementation processes. These solutions often include ready-to-deploy frameworks, automated policy enforcement tools, and centralized management platforms designed to simplify adopting and managing the principles of ZTA. Leveraging these pre-built solutions can expedite the implementation of ZTA, enabling organizations to enhance their cloud security posture efficiently and effectively.

Table 4.2 outlines which vendors solve the specific security aspect of AWS. This section explores different ZTA solutions by various vendors for AWS.

### Microsoft

Microsoft [155] offers a comprehensive suite of solutions for implementing a ZTA, which consists of six key pillars: identity, endpoints, applications, network, infras-

**Table 4.2:** ZTA Solutions Offered by Different Vendors for AWS

| Vendor | Data Security | IAM | Multi-tenancy | Virtualization Security | Network Security | Web & API Security | Visibility |
|---|---|---|---|---|---|---|---|
| Microsoft | ✓ | ✓ | | ✓ | ✓ | ✓ | ✓ |
| VMWare | ✓ | | ✓ | ✓ | ✓ | ✓ | ✓ |
| Cisco | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Zscaler | ✓ | | | | ✓ | ✓ | ✓ |
| Cloudflare | | ✓ | | | ✓ | | ✓ |
| Akamai | | ✓ | | | ✓ | | ✓ |
| Fortinet | | ✓ | | ✓ | ✓ | | ✓ |
| Crowdstrike | | ✓ | | ✓ | | | ✓ |

tructure, and data. In the realm of identity, Microsoft's Entra platform provides a robust set of tools for identity protection. These IAM, Single Sign-On (SSO), MFA, SSE, cloud identity management, and application identity management. Endpoint security is addressed through the utilization of Microsoft Defender and Microsoft Intune. These tools ensure device security, endpoint management, and analytics across various platforms. For secure applications, networks, and infrastructure, Microsoft leverages its Defender suite for applications and cloud apps alongside Entra for permission management. Azure is crucial in network security, facilitating network connections, segmentation, containment, governance, and application management. In the data pillar, Microsoft Purview offers robust security measures, including data management, data loss prevention, and data protection, ensuring comprehensive protection of sensitive data assets.

### VMware

VMware's [156] set of ZTA tools are structured under five fundamental pillars: devices, users, transport/session, application, and data:

- **Devices:** VMware provides Workspace ONE UEM for efficient device management and inventory tracking. VMware Unified Access Gateway plays a pivotal role in device authentication, while VMware Carbon Black and Workspace ONE Mobile Threat Defense ensure robust endpoint protection.
- **Users:** VMware provides ONE Access and ONE Intelligence, facilitating passwordless user authentication, dynamic conditional access, and user risk scoring to enhance security.
- **Transport/Session:** VMware provides Unified Access Gateway and VMware Horizon to establish secure session transport through encryption. VMware NSX-T Data Center complements this by providing least privileged access and micro-segmentation, bolstering network security.

– **Application:** VMware leverages Horizon and Workspace ONE UEM to fortify application security and trust. Workspace ONE Access further enhances security by enabling SSO with applications.

– **Data:** VMware leverages Horizon, Workspace ONE UEM, and NSX-T Data Center to classify data, implement data loss prevention measures, and enforce secure access to data at rest, ensuring comprehensive data security across the organization's infrastructure.

### Cisco

Cisco [157] offers comprehensive zero-trust security solutions that revolve around four key functional requirements: establishing trust, enforcing trust-based access, continuously verifying trust, and responding promptly to changes in trust levels. To implement this model effectively, Cisco provides various tools to enhance user, device, network, and cloud security.

For user and device security, Cisco employs solutions like Cisco DUO and Cisco Cyber Vision. Cisco DUO focuses on MFA and secure access, ensuring that only verified users and devices can access critical resources. Cisco Cyber Vision monitors industrial control systems and critical infrastructure, providing visibility and security for operational technology environments.

In network and cloud security, Cisco offers tools such as Cisco Umbrella and Cisco Secure Workload. Cisco Umbrella provides a secure internet gateway, enforcing security protocols across networks and cloud environments to mitigate data breaches and unauthorized access risks. On the other hand, Cisco Secure Workload ensures comprehensive application security measures by monitoring application behaviors and enforcing security policies to protect against potential vulnerabilities and cyber threats.

### Zscaler

Zscaler [158] offers a cloud-native solution called the "Zero Trust Exchange" that transforms connectivity by focusing on secure access to applications rather than traditional network-based approaches. This platform prioritizes data protection and addresses critical security concerns in the modern digital landscape.

When Zero Trust Exchange integrates with AWS, it ensures comprehensive coverage across various aspects of zero trust architecture, catering to diverse organizational needs and enhancing overall security posture. The Zscaler Zero Trust Exchange reduces risk across all four stages of the attack chain: minimizes the attack surface, prevents compromise, eliminates lateral movement, and stops data loss.

Zscaler's Zero Trust Exchange includes tools designed to implement effective zero trust architectures. These tools provide secure access to applications, enforce strict access controls, and continuously monitor user and device behaviors to detect and mitigate potential threats. By leveraging its expansive partner network, Zscaler offers organizations a versatile and comprehensive suite of solutions that address evolving security challenges and reinforce security measures, bolstering resilience against emerging threats.

### Cloudflare

Cloudflare [159] provides a cloud-native solution for ZTNA, known as Zero Trust as-a-service. This platform operates as a cloud-based service, enhancing access control by leveraging user identity and context, complemented by robust MFA that is resistant to phishing attacks. Unlike traditional VPNs, Cloudflare's ZTNA service aims to enhance security while minimizing the workflow disruptions often associated with conventional ZTA implementations. By harnessing cloud technology, Cloudflare aims to optimize operational efficiency and productivity, offering a secure and flexible network access solution suitable for today's dynamic digital landscape.

### Akamai

Akamai [160] offers a diverse suite of tools designed to support the implementation of a ZTA, providing robust security solutions across multiple domains. One of their key tools, Akamai Guardicore Segmentation, enables effective network segmentation for both on-premise and cloud environments. This solution enhances security posture by isolating critical assets and mitigating potential risks associated with lateral movement within networks. Additionally, Akamai MFA plays a crucial role in identity security, strengthening authentication mechanisms to prevent unauthorized access. Akamai complements these security offerings with Hunt, an advanced threat intelligence and remediation tool that proactively identifies and addresses security threats. This comprehensive approach ensures organizations are well-equipped to defend against evolving cyber threats, fostering a secure operational environment. Beyond tool provision, Akamai supports organizations with comprehensive guides and resources tailored to assist in developing robust ZTA, emphasizing their commitment to empowering organizations with the knowledge and tools necessary for resilient security frameworks in today's dynamic threat landscape.

### Fortinet

Fortinet [161] offers a trio of primary zero trust security solutions aimed at enhancing organizational defenses against evolving cyber threats. FortiClient, a central component in Fortinet's offerings, facilitates ZTNA to ensure secure connectivity while providing robust endpoint protection. Integrated with next-generation antivirus, application firewalls, and endpoint quarantine capabilities, FortiClient delivers comprehensive security measures to safeguard endpoints from various threats. Complementing FortiClient, Fortinet provides FortiToken and FortiAuthenticator, essential tools for effective network access management and SSO capabilities. These solutions strengthen access controls, streamline authentication processes, and enhance overall security posture within organizations. By leveraging Fortinet's zero trust security solutions, organizations can establish a comprehensive security framework encompassing network and endpoint security. This approach enables effective risk mitigation, protection of critical assets, and resilience against evolving cyber threats in today's dynamic threat landscape.

#### CrowdStrike

CrowdStrike [162], a prominent member of The Spectra Alliance, collaborates with leading cybersecurity firms to effectively integrate their ZTA solutions. Within this alliance, CrowdStrike offers a suite of tools to enhance cybersecurity defenses. Their cloud-native zero trust platform stands out, providing organizations with a scalable framework to bolster security posture. The key features include AI-driven identity management and segmentation, which automate critical processes and enhance protection against emerging threats. CrowdStrike's expertise in threat intelligence enables proactive monitoring and analysis, helping organizations stay ahead of evolving cyber threats. Their endpoint security solutions ensure comprehensive protection, safeguarding endpoints from malicious activities and unauthorized access attempts.

This chapter discussed the security challenges present in AWS environments and how ZTA can enhance AWS security. Furthermore, we explored the tools offered by AWS and solutions offered by third-party vendors to implement ZTA in AWS environments. The next chapter will present a POC implementation of ZTA to securely manage AWS resources.

# Chapter 5
# Proof-of-Concept Implementation

In chapter 2, we discussed the services offered by AWS, and in the previous chapter 4, we discussed different security challenges in AWS environments and how to tackle them through ZTA. Though AWS offers solutions for deploying and managing resources such as VMs and databases, the challenge lies in maintaining a secure, scalable, and easy-to-manage network that meets the organization's requirements.

This chapter describes a POC implementation of a ZTA system with Tailscale to securely manage RDS and VMs on AWS, leveraging the capabilities of Tailscale to create a private, encrypted, and authenticated network. The implementation in this thesis aims to serve as a POC for the application of ZTA at Sikt - the Norwegian Infrastructure Provider for Research and Education in Norway.

This chapter is organized as follows. We describe the status quo and requirements for a secure system to manage AWS resources at Sikt. Next, we detail the existing methods and identify the corresponding issues. We then discuss how ZTA principles can address these issues. Following this, we provide an overview of the POC implementation and an in-depth description of various components within our zero-trust system. Finally, we outline the challenges with the implementation and discuss how this implementation can be adopted at Sikt or other organizations.

## 5.1 Challenge in AWS Resource Management

Sikt provides various services for Norwegian educational and research institutions. Examples include Feide, StudentWeb, UNINETT compute services, etc. Sikt is organized into product areas, which comprise several product teams that develop and maintain the products. Platform teams provide the development infrastructure to these product teams in Sikt. AWS is one of the main cloud providers used by various teams across Sikt. The platform team manages and provides AWS accounts and resources to product teams within Sikt.

The platform team usually provides AWS accounts to the product teams for their development work. Often, these teams manage their own resources in AWS accounts. However, not all teams have the necessary resources and skills to maintain their AWS accounts. In such cases, the platform team takes responsibility for creating and managing resources for those teams. These resources typically include VMs (EC2) and RDS instances.

Considering the management overheads associated with managing several resources in multiple AWS accounts, the platform team creates and manages all the resources in one or two AWS accounts. To grant access to these resources, the platform team opens Secure Shell (SSH) ports for the instances and provides access keys or credentials to the respective teams. The credentials for SSH access are created once and are directly provided to the teams as needed.

There are several challenges with this kind of setup. Managing SSH ports, access keys, and credentials is error-prone and can lead to security vulnerabilities. If one set of credentials is compromised, it can potentially expose all resources in the account. Manually managing access becomes increasingly inefficient as the number of teams and resources grows. Furthermore, opening SSH ports to provide access increases the attack surface, making instances more vulnerable to unauthorized access and attacks.

The current way to ensure that only authorized people access their resources is to use Bastion Host with Internet Protocol (IP) filtering. A bastion host is set up with a security group allowing inbound SSH access only from trusted IP addresses. The resource instances are configured only to allow SSH access from the bastion host's IP address. However, there are several potential shortcomings:

- RDS instances have dynamic IP addresses, so it is difficult to use static IP addresses and use IP filtering.
- IP addresses can be easily spoofed.
- If any resource is compromised, the attacker can use that as a jump host or proxy to attack all the other resources.
- Bastion host becomes a single point of failure. If it goes down, access to all other resource instances is blocked.

The principles of ZTA can be applied to solve various issues listed above. For example, since ZTA focuses on active verification of user identity, passive IP-based filtering or access will be replaced by active user identification in a ZTA. IP spoofing can be mitigated since access decisions are based on user identity, device health, and behavior analytics rather than IP addresses alone.

Furthermore, ZTA would divide the network into granular zones, each with its own security policies isolated from other zones. Even if an attacker gains access to one zone, they cannot easily move laterally to other zones. Ensuring that users and machines have only the minimum necessary access to perform their tasks reduces the impact of a compromised resource instance. Continuous monitoring, an important facet of a ZTA, can detect anomalies in behavior that could indicate a compromised instance. Adaptive policies can adjust access permissions in real time based on observed risks.

A full-fledged implementation of ZTA is outside the scope of this master's thesis. However, a POC was implemented to demonstrate ZTA's capability in solving the challenges mentioned above. The rest of this chapter is structured as follows: We first discuss the POC implementation, its intricacies, and challenges. Then, we discuss how this prototype can be adopted to solve the challenges listed in the current setup.

## 5.2 Overview of the Implementation

The implementation is a prototype of a Zero Trust system designed to manage access to AWS resources, including VMs and RDS. Figure 5.1 provides the overview of the setup. The main components are:

- Nodes: A node is a combination of a user and a device that can be physical or virtual. A node requests access to the AWS resources.
- Subnet Router: The subnet router routes the traffic securely to the AWS resources from the nodes. It acts as the PEP as shown in the Figure 2.1 in Chapter 2. In this case, it is an EC2 machine with tailscale configured on it.
- Resources: These are the resources that are being accessed on AWS. These include EC2, RDS, S3 etc.
- Tailscale Client and Coordination Server: The tailscale client and coordination server combine together to act as the PDP as shown in the Figure 2.1 in Chapter 2.
    - Tailscale Client: The Tailscale client is an interface accessible to the administrator. It allows the admin to smoothly manage the access control policies. It acts as the PA as shown in the Figure 2.1 in Chapter 2.
    - Tailscale Coordination Server: The coordination server is a central server that maintains a connection to all machines in the network. It manages

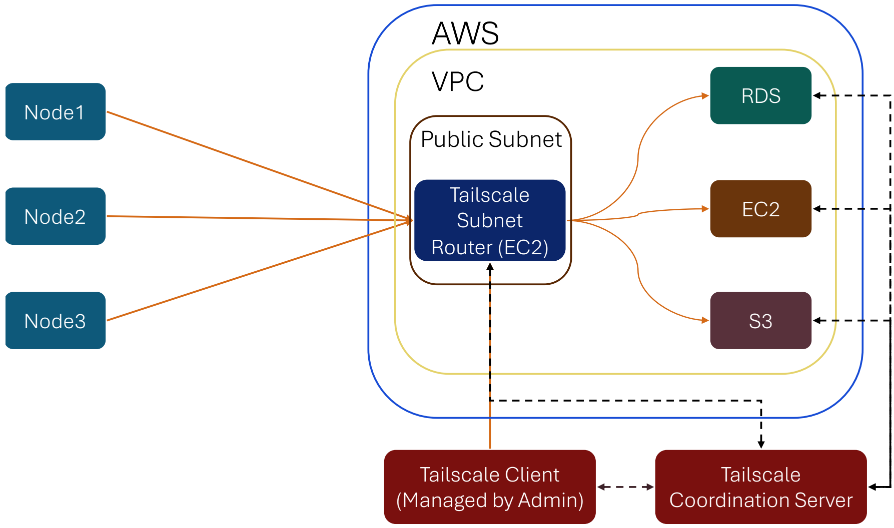



**Figure 5.1:** POC Implementation of ZTA with Tailscale

encryption keys, network changes, access policy changes, and maintains a connection to all machines in the network. It acts as the PE as shown in the Figure 2.1 in Chapter 2.

**Workflow**: In this implementation, nodes request access to resources hosted on AWS. The access request is sent to the subnet router, which is the funnel for the incoming connections and routes the traffic to the designated AWS resources. The subnet router prompts the nodes to prove their identity. Identity verification can be done via a username-password scheme or a popular identity provider like Microsoft. The administrator sets up the type of identity verification. An advantage of using identity providers is that they have inbuilt MFA setups through authenticator applications on smartphones or SMS-based OTP.

After the identity is proven, the access requests are routed through the subnet router (an EC2 instance configured with Tailscale). This EC2 instance resides within a public subnet, which is part of a VPC on AWS. The policy or access decisions made on this router are configured via the Tailscale client.

Tailscale Client provides the administrator with an interface that provides granular access control. These policy changes are enforced on the resources through the

Tailscale coordination server. Once the router receives the information that the node is allowed access to the resource it is requesting access to, the router provides the connection between the node and the resource, ensuring only authorized nodes can access the specified resources with a secure connection.

The enforcement of ZTA principles is abstracted away from this whole system and is taken care of by tailscale, allowing for a seamless user experience. Tailscale provides the feature of segmenting the network into smaller segments and managing them with separate security policies. Tailscale takes care of making sure the traffic and connection are encrypted. This eliminated the need to open SSH ports for each node. This enhances security by reducing the attack surface and simplifying the network management. In the next subsections, we give an overview of Tailscale and the steps required for the implementation.

### 5.2.1 Overview of Tailscale

Tailscale [163] is a SDN solution that streamlines the process of establishing a secure, private network over the Internet. It leverages the WireGuard VPN protocol, renowned for its robust cryptography and high-performance capabilities. Tailscale's architecture enables devices to connect directly to each other using end-to-end encryption, creating a secure and efficient mesh network. This direct connection approach bypasses the need for centralized servers or complicated configurations, simplifying setup and management.

By adopting the zero trust security model, Tailscale ensures that no device is trusted by default. Instead, rigorous access controls are implemented, requiring authentication and authorization for each connection attempt. This model enhances security by minimizing the risk of unauthorized access and ensuring that every device in the network is continuously verified. Tailscale integrates seamlessly with identity providers such as Google Workspace, Microsoft Azure Active Directory, and others, further enhancing security through SSO and MFA.

Additionally, Tailscale manages network complexities like Network Address Translation (NAT) traversal and dynamic IP addresses automatically, making it an ideal solution for both personal and enterprise environments. Its user-friendly interface allows for easy administration and monitoring of network connections, providing a scalable solution that can accommodate a wide range of devices and use cases, from simple remote access to complex corporate networks.

**Tailnet:** A tailnet functions as a private network within Tailscale. Upon initial login to Tailscale on a phone, laptop, desktop, or cloud VM, a tailnet is automatically established. Under the basic plan (personal plan), users can connect multiple devices and accommodate up to three users within their tailnet. Each device within the tailnet is assigned a private Tailscale IP address within the Carrier-Grade Network Address Translation (CGNAT) range, enabling direct communication between devices across the internet. For businesses, organizations, and other plan users, tailnets can scale to support numerous devices and users. These tailnets can be integrated with Microsoft Active Directory, Google Workspace, GitHub organizations, Okta tenancies, or other identity provider namespaces as required.

### 5.2.2 Implementation Steps

**Prerequities:** To utilize Tailscale, one must first register for an account [164], selecting from various plans that cater to different numbers of users and devices based on specific needs. Typically, enterprises opt for the enterprise version to leverage all available benefits. However, in the context of this implementation, the basic version sufficed. Following registration, access to the Tailscale client allows users to set up a tailnet. The Tailscale admin console displays all users and devices, managed by the administrator responsible for the configuration.

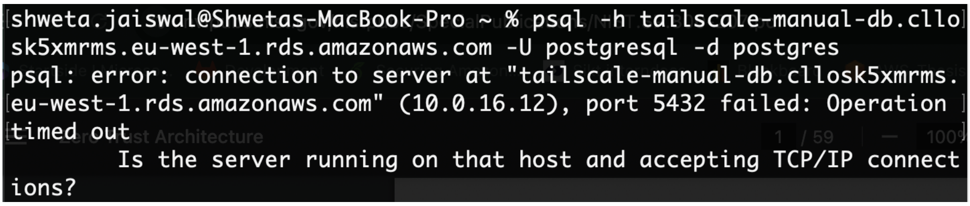
**Node setup:** To set up nodes, the Tailscale client is downloaded and installed on various devices. Once installed, users sign up using their existing personal plan account or log in using the designated enterprise identity provider for enterprise plans. Following login, each device requests authorization to join the tailnet. Admin approval through the console is required to finalize device inclusion into the tailnet.

**Setting Up EC2 Instance as Subnet Router:** The POC implementation dealt with secure access to various RDS instances within the same AWS account. To achieve this, we configured an EC2 instance as a subnet router within the AWS environment. The following steps were undertaken to set up the EC2 instance as a subnet router:

1. **EC2 Instance Creation**: Initially, an EC2 instance was created in the AWS account. This can be done manually through the AWS Management Console or using Infrastructure-as-Code (IaC) tools like CloudFormation, Terraform, or Pulumi. In our case, Pulumi was utilized to automate infrastructure provisioning.

2. **Operating System Selection**: The EC2 instance was launched with the Amazon Linux operating system, suitable for the intended network routing tasks.

3. **Tailscale Installation**: Upon provisioning the EC2 instance, Tailscale Linux packages were installed. These packages are necessary for integrating the EC2 instance into the Tailscale network.



```
shweta.jaiswal@Shwetas-MacBook-Pro ~ % psql -h tailscale-manual-db.cllo
sk5xmrms.eu-west-1.rds.amazonaws.com -U postgresql -d postgres
psql: error: connection to server at "tailscale-manual-db.cllosk5xmrms.
eu-west-1.rds.amazonaws.com" (10.0.16.12), port 5432 failed: Operation
timed out
        Is the server running on that host and accepting TCP/IP connect
ions?
```

**Figure 5.2:** Access request when not on the Tailscale network

4. **Configuration of Subnet Routing**: Subsequently, subnet routing configuration was set up on the EC2 instance. This involved configuring network routing tables and policies to facilitate secure communication between the EC2 instance acting as the subnet router and the various RDS instances within the AWS VPC.

5. **Security Group Updates**: The security policies of the RDS instances were updated to allow inbound connections from the subnet router EC2 instance. This step ensures that the subnet router can securely communicate with the RDS instances.

6. **DNS Setup**: To enable seamless access using AWS DNS names within the network, split DNS setup was configured. This involves specifying DNS server settings in the Tailscale admin console and restricting DNS queries to the specific domain associated with AWS resources.

7. **Security Group Adjustment**: With private SSH access established via the Tailscale network, public SSH access rules were removed from the EC2 instance's security group. This enhances security by limiting access to the EC2 instance to internal network connections only.

8. **Network Connectivity**: Following setup completion, connectivity tests were conducted to ensure that all configured components, including the EC2 subnet router, RDS instances, and Tailscale network, functioned correctly and securely.

### 5.2.3 Testing of the Implementation

To assess the integrity of the POC implementation, we conducted a series of tests to evaluate access to the RDS instance under different scenarios.

- **Access from Outside the Tailscale Network:** When the device was not connected to the Tailscale network configured for the secure access of the resource, the connection request timed out. This occurred because the device was not connected to the subnet on which the RDS instance resides (Figure 5.2).

```
shweta.jaiswal@Shwetas-MacBook-Pro ~ % psql -h tailscale-manual-db.cllo
sk5xmrms.eu-west-1.rds.amazonaws.com -U postgresql -d postgres
Password for user postgresql:
psql: error: connection to server at "tailscale-manual-db.cllosk5xmrms.
eu-west-1.rds.amazonaws.com" (10.0.16.12), port 5432 failed: FATAL:  pa
ssword authentication failed for user "postgresql"
connection to server at "tailscale-manual-db.cllosk5xmrms.eu-west-1.rds
.amazonaws.com" (10.0.16.12), port 5432 failed: FATAL:  no pg_hba.conf
entry for host "10.0.8.96", user "postgresql", database "postgres", no
```

**Figure 5.3:** Access request when on Tailscale network but with incorrect credentials

```
shweta.jaiswal@Shwetas-MacBook-Pro ~ % psql -h tailscale-manual-db.cllo
sk5xmrms.eu-west-1.rds.amazonaws.com -U postgresql -d postgres
Password for user postgresql:
psql (14.11 (Homebrew), server 16.1)
WARNING: psql major version 14, server major version 16.
         Some psql features might not work.
SSL connection (protocol: TLSv1.3, cipher: TLS_AES_256_GCM_SHA384, bits
: 256, compression: off)
Type "help" for help.

postgres=>
```

**Figure 5.4:** Access request successful in connecting to the RDS instance

– **Access from the Correct Subnet with Incorrect Credentials**: When the user was on the correct subnet, the same as the RDS instance, but provided incorrect credentials, access was denied. This is depicted in Figure 5.3.

– **Access from the Correct Subnet with Correct Credentials**: When the user was on the correct subnet, the same as the RDS instance, and provided the correct credentials, the user was authenticated. The user was then granted access to the RDS instance, allowing them to access the database and perform queries as needed (Figure 5.4).

## 5.3 Evaluation of the Implementation

In this section, we qualitatively evaluate the POC implementation that relies on the ZTA solution provided by Tailscale using factors including ease of setup, cost, dependency, reliance, and trust.

### 5.3.1 Ease of Setup

Tailscale has demonstrated notable ease of setup in this implementation. The process involved installing the Tailscale client on the subnet router and adding users with devices to the subnet via the Tailscale admin client. Tailscale's coordination servers autonomously managed the network configuration, significantly reducing the networking overhead.

### 5.3.2 Cost of Integration

In this implementation, the free-tier version of Tailscale was utilized, allowing only three users to be added. However, the business or enterprise version incurs costs, especially as the number of nodes increases. This requires balancing the need for ZTA with budget considerations for a solution like Tailscale.

### 5.3.3 Dependency on Administrator for Access

Tailscale's access control requires an administrator to manage and grant access permissions. This centralization can become a bottleneck, particularly in larger organizations where numerous users need onboarding or frequent updates to their access permissions.

### 5.3.4 Reliance on Coordination Server

The Tailscale coordination server is easy to set up, so users do not need to manage the network. However, if Tailscale's coordination servers experience downtime or technical issues, network connectivity can be disrupted, affecting resource access.

### 5.3.5 One-Account One-Admin per Role

Tailscale allows for multiple administrators, each managing different areas (e.g., billing, network), but it does not permit multiple administrators for a single area. This limitation complicates management, as only one person can be designated admin per area. Alternatively, multiple platform engineers would need to share admin account credentials to configure various policies, which is inefficient and risky. While creating multiple Tailscale accounts could address this, it adds complexity and cost to overall management.

### 5.3.6 Literacy Among Developers

Regarding networking and setup, Tailscale abstracts away all the complexities. However, developers need to have a basic understanding of networking concepts, and there might be a slight learning curve to understand Tailscale's unique approach to mesh-based networks.

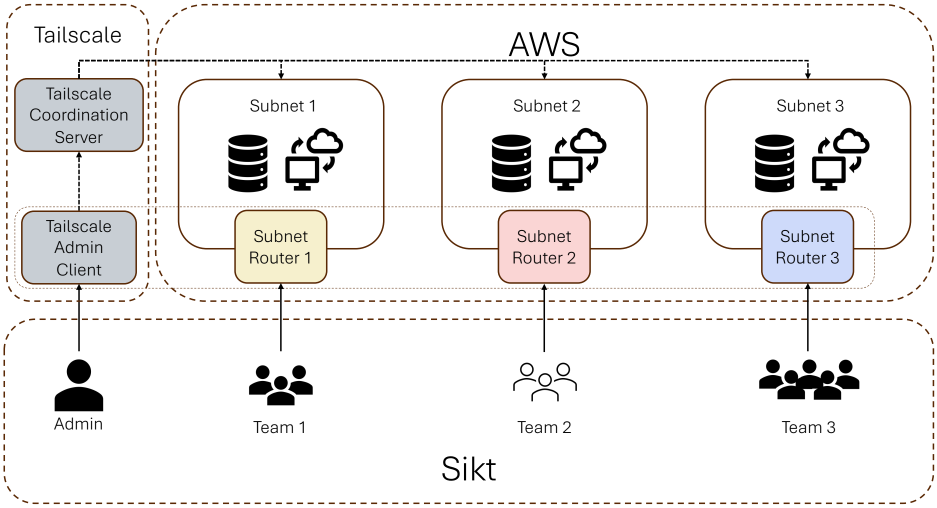



**Figure 5.5:** Demonstration of POC Implementation in Sikt

### 5.3.7 Trust in Third-Party Provider

Configuring the subnet router necessitates root access on the host machine, implying that developers must entrust Tailscale with such elevated privileges. In addition, developers need confidence in the reliability and minimal downtime of Tailscale's coordination servers. Dependence on a third-party provider like Tailscale may raise concerns about vendor lock-in. Hence, organizations should assess the long-term effects of relying on Tailscale and consider potential exit strategies if necessary.

## 5.4 Adoption in a Real-World Use Case

This implementation can be adopted by Sikt to address the challenges described in Section 5.1. A similar strategy can be employed by other organizations facing analogous challenges in managing AWS resources. In this section, we discuss how each component of the POC implementation would be applied in a real-world scenario of ZTA at Sikt for AWS resource management. Figure 5.5 provides an demonstration on how POC implementation can be adopted sat Sikt.

- Users and their devices, attempting to access AWS resources, will function as nodes within the network.

– For each team, a subnet router can be deployed on an EC2 instance, with each subnet router adhering to its security policy. Implementing separate subnet routers for each team will facilitate the creation of isolated networks, ensuring team-specific network segregation and enhanced security.

– The platform team will assume the administration role, configuring appropriate access to resources through the access control list in the Tailscale admin client. This central administration will streamline resource access management and ensure compliance with security protocols.

– The security policy for AWS resources, such as RDS or EC2 instances belonging to a particular team, will be configured to reside within that team's isolated network. This will ensure that resources are secured according to each team's specific needs and policies.

This chapter explained the implementation of the POC, its steps, and tests in different scenarios. This chapter evaluated the implementation and discussed how the POC implementation can be adopted in a full-fledged ZTA system.

# Chapter 6 Conclusion and Future Work

This chapter provides a conclusion and future directions for the work. Finally, it contextualizes this thesis within the UN SDGs.

## 6.1 Conclusion

In conclusion, this thesis provided a comprehensive exploration into using ZTA to enhance the security of AWS environments. It offered a detailed review of state-of-the-art ZTA techniques, highlighted the security challenges inherent in AWS, and discussed ways to address these challenges through ZTA principles. Furthermore, it presented a POC implementation of a zero-trust system within an AWS environment. The thesis underscored the urgent need to move away from traditional security measures towards a more advanced security model in the face of growing cyber threats. It pointed out that ZTA's adaptive and dynamic approach to security, which includes continuous authentication and authorization, could effectively mitigate risks associated with credential theft, privilege escalation, and insider threats.

The findings of this study provide valuable insights for organizations looking to strengthen their cloud infrastructure security, particularly those utilizing AWS. Ultimately, applying ZTA principles in AWS environments could lead to more secure, robust, and resilient cloud infrastructures.

## 6.2 Future Work

Several areas of exploration emerge from this thesis for future work. Firstly, an examination of ZTA implementation in other cloud environments, such as GCP and Microsoft Azure, could be undertaken. Further investigation could be conducted in hybrid or multi-cloud environments. This would allow for a comparative study on the effectiveness of ZTA across different cloud platforms.

Secondly, while the POC implementation has proven valuable, more extensive testing and evaluation in a real-world setting would provide more practical and robust insights. Future research could explore how ZTA techniques can be scaled and adapted for larger, more complex AWS environments.

Thirdly, in light of the evolving nature of cyber threats, continuous updates and advancements in ZTA techniques are necessary. Future work could focus on developing more advanced ZTA techniques, particularly in the area of AI and machine learning, to improve threat detection and response times.

Lastly, a more detailed study could be conducted on the economic impact of implementing ZTA in AWS environments. This would give organizations a clearer understanding of the cost-benefit ratio of adopting ZTA principles, a crucial factor in decision-making processes.

In conclusion, the future of ZTA in AWS environments holds immense potential, and further research in the outlined areas would significantly contribute to the field.

## 6.3 United Nations' Sustainable Development Goals

This thesis focuses on securing AWS using ZTA, which aligns with several UN SDGs by enhancing digital infrastructure security and ensuring the protection of data and privacy. This section outlines how our research contributes to specific SDGs and their sub-goals.

### 6.3.1 SDG 9

Goal 9 aims to build resilient infrastructure, promote inclusive and sustainable industrialization, and foster innovation. Within this goal, the most relevant target for our research is target 9.1 - "Develop quality, reliable, sustainable and resilient infrastructure, including regional and transborder infrastructure, to support economic development and human well-being, with a focus on affordable and equitable access for all".

This thesis contributes to developing a more secure and resilient cloud infrastructure by discussing and implementing the ZTA in AWS. This enhances the reliability and sustainability of the digital infrastructure, which is crucial for economic development and the well-being of societies that rely on cloud services.

Securing cloud environments encourages organizations to adopt and innovate using cloud technologies, knowing that their data and operations are protected against cyber threats.

### 6.3.2 SDG 11

Goal 11 focuses on making cities and human settlements inclusive, safe, resilient, and sustainable. While not immediately apparent, there is a relevant target, target 11.7 - "Provide universal access to safe, inclusive and accessible, green and public spaces, particularly for women and children, older persons and persons with disabilities."

A secure cloud infrastructure is essential for smart city initiatives that rely on digital solutions to manage urban resources efficiently. By securing AWS, our research indirectly supports the safe deployment of digital public spaces and services, which are integral to the development of sustainable and resilient urban environments.

### 6.3.3 SDG 16

Goal 16 aims to promote peaceful and inclusive societies, provide access to justice for all, and build effective, accountable, and inclusive institutions at all levels. A key target relevant to our research is target 16.10 - "Ensure public access to information and protect fundamental freedoms, in accordance with national legislation and international agreements"

ZTA emphasizes strict verification processes and continuous monitoring, ensuring that data stored in AWS is secure from unauthorized access. This supports the protection of fundamental freedoms related to privacy and the right to information.

Our research enhances security measures, contributing to building trust in digital infrastructures and promoting transparency and accountability in data management and access.

In summary, securing AWS with ZTA contributes to several UN SDGs by enhancing the security and resilience of digital infrastructures, protecting data and privacy, and fostering innovation. By aligning our research with these global targets, we demonstrate how advanced cybersecurity measures can support broader efforts to achieve a sustainable and prosperous future for all.

# References

[1] AWS Market-Share. URL: https://www.srgresearch.com/articles/cloud-market-gets-its-mojo-back-q4-increase-in-cloud-spending-reaches-new-highs (last visited: Mar. 24, 2024).

[2] AWS Services. URL: https://aws.amazon.com/types-of-cloud-computing/ (last visited: Mar. 24, 2024).

[3] James Guffey and Yanyan Li. «Cloud Service Misconfigurations: Emerging Threats, Enterprise Data Breaches and Solutions». In: *2023 IEEE 13th Annual Computing and Communication Workshop and Conference (CCWC)*. IEEE. 2023, pp. 0806–0812.

[4] Shaharyar Khan, Ilya Kabanov, et al. «A systematic analysis of the capital one data breach: Critical lessons learned». In: *ACM Transactions on Privacy and Security* 26.1 (2022), pp. 1–29.

[5] Imperva Data Breach. URL: https://www.imperva.com/blog/ceoblog/ (last visited: Apr. 14, 2024).

[6] VA Stafford. «Zero trust architecture». In: *NIST special publication* 800 (2020), p. 207.

[7] Nardine Basta, Muhammad Ikram, et al. «Towards a zero-trust micro-segmentation network security strategy: an evaluation framework». In: *NOMS 2022-2022 IEEE/IFIP Network Operations and Management Symposium*. IEEE. 2022, pp. 1–7.

[8] Romans Vanickis, Paul Jacob, et al. «Access control policy enforcement for zero-trust-networking». In: *2018 29th Irish Signals and Systems Conference (ISSC)*. IEEE. 2018, pp. 1–6.

[9] Qigui Yao, Qi Wang, et al. «Dynamic access control and authorization system based on zero-trust architecture». In: *Proceedings of the 2020 1st international conference on control, robotics and intelligent system*. 2020, pp. 123–127.

[10] Kee Hock Tan. «Mitigating Insider Threats in AWS: A Zero Trust Perspective». In: (2023).

[11] Jayachander Surbiryala and Chunming Rong. «Cloud computing: History and overview». In: *2019 IEEE Cloud Summit*. IEEE. 2019, pp. 1–7.

[12] Ali Sunyaev and Ali Sunyaev. «Cloud computing». In: *Internet computing: Principles of distributed systems and emerging internet-based technologies* (2020), pp. 195–236.

[13] AWS Locations. URL: https://www.cloudzero.com/blog/aws-data-center-locations/#:~:text=AWS%20data%20centers%20are%20spread,and%20115%20Direct%20Connect%20Locations. (last visited: Mar. 24, 2024).

[14] Stephen Northcutt, Lenny Zeltser, et al. *Inside Network Perimeter Security (Inside)*. Sams, 2005.

[15] Guo Ya-Jun, Hong Fan, et al. «An access control model for ubiquitous computing application». In: (2005).

[16] Rory Ward and Betsy Beyer. «Beyondcorp: A new approach to enterprise security». In: *; login:: the magazine of USENIX & SAGE* 39.6 (2014), pp. 6–11.

[17] Barclay Osborn. «Beyondcorp: Design to deployment at google». In: *Usenix*. Vol. 41. 2016, p. 28.

[18] VMware NSX. URL: https://www.vmware.com/content/dam/digitalmarketing/vmware/en/pdf/solutions/vmware-network-microsegmentation-solution-overview.pdf (last visited: May 24, 2024).

[19] Sreejith Keeriyattil and Sreejith Keeriyattil. «Microsegmentation and zero trust: Introduction». In: *Zero Trust Networks with VMware NSX: Build Highly Secure Network Architectures for Your Data Centers* (2019), pp. 17–31.

[20] Forrester's ZTX. URL: https://www.forrester.com/blogs/category/zero-trust-security-framework-ztx/ (last visited: May 24, 2024).

[21] ZTNA. URL: https://www.securityweek.com/working-securely-anywhere-zero-trust/ (last visited: May 24, 2024).

[22] MFA. URL: https://www.rsa.com/resources/blog/multi-factor-authentication/what-is-mfa/ (last visited: Apr. 6, 2024).

[23] RSA. URL: https://www.rsa.com/ (last visited: Apr. 6, 2024).

[24] YubiKey. URL: https://www.yubico.com/ (last visited: Apr. 6, 2024).

[25] S Hemalatha. «A systematic review on Fingerprint based Biometric Authentication System». In: *2020 International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE)*. IEEE. 2020, pp. 1–4.

[26] Javier Jose Diaz Rivera, Afaq Muhammad, and Wang-Cheol Song. «Securing Digital Identity in the Zero Trust Architecture: A Blockchain Approach to Privacy-Focused Multi-Factor Authentication». In: *IEEE Open Journal of the Communications Society* (2024).

[27] Divya Sivasankaran, Mona Ragab, et al. «Context-aware fusion for continuous biometric authentication». In: *2018 International Conference on Biometrics (ICB)*. IEEE. 2018, pp. 233–240.

[28] Syed Wajid Ali Shah, Naeem Firdous Syed, et al. «Towards a lightweight continuous authentication protocol for device-to-device communication». In: *2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*. IEEE. 2020, pp. 1119–1126.

[29] Syed W Shah, Naeem F Syed, et al. «LCDA: Lightweight continuous device-to-device authentication for a zero trust architecture (ZTA)». In: *Computers & Security* 108 (2021), p. 102351.

[30] Lei Meng, Daochao Huang, et al. «A continuous authentication protocol without trust authority for zero trust architecture». In: *China Communications* 19.8 (2022), pp. 198–213.

[31] Iurii Matiushin and Vladimir Korkhov. «Continuous Authentication Methods for Zero-Trust Cybersecurity Architecture». In: *International Conference on Computational Science and Its Applications*. Springer. 2023, pp. 334–351.

[32] Ali Abdullah S AlQahtani, Zakaria El-Awadi, and Manki Min. «A survey on user authentication factors». In: *2021 IEEE 12th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*. IEEE. 2021, pp. 0323–0328.

[33] Eiji Hayashi, Sauvik Das, et al. «Casa: context-aware scalable authentication». In: *Proceedings of the Ninth Symposium on Usable Privacy and Security*. 2013, pp. 1–10.

[34] Kamal Benzekki, Abdeslam El Fergougui, and Abdelbaki ElBelrhiti ElAlaoui. «A context-aware authentication system for mobile cloud computing». In: *Procedia Computer Science* 127 (2018), pp. 379–387.

[35] Yosef Ashibani, Dylan Kauling, and Qusay H Mahmoud. «Design and implementation of a contextual-based continuous authentication framework for smart homes». In: *Applied System Innovation* 2.1 (2019), p. 4.

[36] Khairul Azmi Abu Bakar and Galoh Rashidah Haron. «Adaptive authentication: Issues and challenges». In: *2013 World Congress on Computer and Information Technology (WCCIT)*. IEEE. 2013, pp. 1–6.

[37] Xiangshuai Yan and Huijuan Wang. «Survey on zero-trust network security». In: *Artificial Intelligence and Security: 6th International Conference, ICAIS 2020, Hohhot, China, July 17–20, 2020, Proceedings, Part I 6*. Springer. 2020, pp. 50–60.

[38] Songpon Teerakanok, Tetsutaro Uehara, and Atsuo Inomata. «Migrating to zero trust architecture: Reviews and challenges». In: *Security and Communication Networks* 2021.1 (2021), p. 9947347.

[39] Christoph Buck, Christian Olenberger, et al. «Never trust, always verify: A multivocal literature review on current knowledge and research gaps of zero-trust». In: *Computers & Security* 110 (2021), p. 102436.

[40] Sirshak Sarkar, Gaurav Choudhary, et al. «Security of zero trust networks in cloud computing: A comparative review». In: *Sustainability* 14.18 (2022), p. 11213.

[41] Farhan A Qazi. «Study of zero trust architecture for applications and network security». In: *2022 IEEE 19th International Conference on Smart Communities: Improving Quality of Life Using ICT, IoT and AI (HONET)*. IEEE. 2022, pp. 111–116.

[42] Naeem Firdous Syed, Syed W Shah, et al. «Zero trust architecture (zta): A comprehensive survey». In: *IEEE Access* 10 (2022), pp. 57143–57179.

[43] Yuanhang He, Daochao Huang, et al. «A survey on zero trust architecture: Challenges and future trends». In: *Wireless Communications and Mobile Computing* 2022.1 (2022), p. 6476274.

[44] Hongzhaoning Kang, Gang Liu, et al. «Theory and Application of Zero Trust Security: A Brief Survey». In: *Entropy* 25.12 (2023), p. 1595.

[45] Poonam Dhiman, Neha Saini, et al. «A Review and Comparative Analysis of Relevant Approaches of Zero Trust Network Model». In: *Sensors* 24.4 (2024), p. 1328.

[46] Eduardo B Fernandez and Andrei Brazhuk. «A critical analysis of Zero Trust Architecture (ZTA)». In: *Computer Standards & Interfaces* 89 (2024), p. 103832.

[47] Abraham Itzhak Weinberg and Kelly Cohen. «Zero Trust Implementation in the Emerging Technologies Era: Survey». In: *arXiv preprint arXiv:2401.09575* (2024).

[48] Cornelius Itodo and Murat Ozer. «Multivocal Literature Review on Zero-Trust Security Implementation». In: *Computers & Security* (2024), p. 103827.

[49] Patricia Arias-Cabarcos, Christian Krupitzer, and Christian Becker. «A survey on adaptive authentication». In: *ACM Computing Surveys (CSUR)* 52.4 (2019), pp. 1–30.

[50] Lewis Golightly, Paolo Modesti, et al. «Securing distributed systems: A survey on access control techniques for cloud, blockchain, IoT and SDN». In: *Cyber Security and Applications* (2023), p. 100015.

[51] Sejong Oh and Seog Park. «Task–role-based access control model». In: *Information systems* 28.6 (2003), pp. 533–562.

[52] Salah T Alshammari, Aiiad Albeshri, and Khalid Alsubhi. «Integrating a high-reliability multicriteria trust evaluation model with task role-based access control for cloud services». In: *Symmetry* 13.3 (2021), p. 492.

[53] Marc-André Laverdière, Karl Julien, and Ettore Merlo. «RBAC protection-impacting changes identification: A case study of the security evolution of two PHP applications». In: *Information and Software Technology* 139 (2021), p. 106630.

[54] Eric Freudenthal, Tracy Pesin, et al. «dRBAC: distributed role-based access control for dynamic coalition environments». In: *Proceedings 22nd International Conference on Distributed Computing Systems*. IEEE. 2002, pp. 411–420.

[55] Youngmin Jung and Mokdong Chung. «Adaptive security management model in the cloud computing environment». In: *2010 the 12th International Conference on Advanced Communication Technology (ICACT)*. Vol. 2. IEEE. 2010, pp. 1664–1669.

[56] Jian Xu, Yanbo Yu, et al. «Role-based access control model for cloud storage using identity-based cryptosystem». In: *Mobile Networks and Applications* 26 (2021), pp. 1475–1492.

[57] Vincent C Hu, David Ferraiolo, et al. «Guide to attribute based access control (abac) definition and considerations (draft)». In: *NIST special publication* 800.162 (2013), pp. 1–54.

[58] Daniel Servos and Sylvia L Osborn. «Current research and open problems in attribute-based access control». In: *ACM Computing Surveys (CSUR)* 49.4 (2017), pp. 1–45.

[59] Yingjie Xue, Jianan Hong, et al. «LABAC: A location-aware attribute-based access control scheme for cloud storage». In: *2016 IEEE Global Communications Conference (GLOBECOM)*. IEEE. 2016, pp. 1–6.

[60] Pedro García-Teodoro, José Camacho, et al. «A novel zero-trust network access control scheme based on the security profile of devices and users». In: *Computer Networks* 212 (2022), p. 109068.

[61] Jiuru Wang, Zhiyuan Wang, et al. «Attribute and User Trust Score-Based Zero Trust Access Control Model in IoV». In: *Electronics* 12.23 (2023), p. 4825.

[62] Khaled Riad, Teng Huang, and Lishan Ke. «A dynamic and hierarchical access control for IoT in multi-authority cloud storage». In: *Journal of Network and Computer Applications* 160 (2020), p. 102633.

[63] I Indu and PM Rubesh Anand. «Identity and access management for cloud web services». In: *2015 IEEE Recent Advances in Intelligent Computational Systems (RAICS)*. IEEE. 2015, pp. 406–410.

[64] Ehtesham Zahoor, Zubaria Asma, and Olivier Perrin. «A formal approach for the verification of AWS IAM access control policies». In: *Service-Oriented and Cloud Computing: 6th IFIP WG 2.14 European Conference, ESOCC 2017, Oslo, Norway, September 27-29, 2017, Proceedings 6*. Springer. 2017, pp. 59–74.

[65] Maanak Gupta, Feras M Awaysheh, et al. «An attribute-based access control for cloud enabled industrial smart vehicles». In: *IEEE Transactions on Industrial Informatics* 17.6 (2020), pp. 4288–4297.

[66] Yang Yang, Yanjiao Chen, et al. «Identity-based cloud storage auditing for data sharing with access control of sensitive information». In: *IEEE Internet of Things Journal* 9.13 (2021), pp. 10434–10445.

[67] Yvo Desmedt and Arash Shaghaghi. «Function-based access control (fbac): Towards preventing insider threats in organizations». In: *From Database to Cyber Security: Essays Dedicated to Sushil Jajodia on the Occasion of His 70th Birthday* (2018), pp. 143–165.

[68] Hany F Atlam, Muhammad Ajmal Azad, et al. «Risk-based access control model: A systematic literature review». In: *Future Internet* 12.6 (2020), p. 103.

[69] Brian Lee, Roman Vanickis, et al. «Situational awareness based risk-adapatable access control in enterprise networks». In: *arXiv preprint arXiv:1710.09696* (2017).

[70] Caixia Yang, Liang Tan, et al. «AuthPrivacyChain: A blockchain-based access control framework with privacy protection in cloud». In: *IEEE Access* 8 (2020), pp. 70604–70615.

[71] Keke Gai, Yufeng She, et al. «A blockchain-based access control scheme for zero trust cross-organizational data sharing». In: *ACM Transactions on Internet Technology* 23.3 (2023), pp. 1–25.

[72] Sheng Ding, Jin Cao, et al. «A novel attribute-based access control scheme using blockchain for IoT». In: *IEEE Access* 7 (2019), pp. 38431–38441.

[73] Samia Masood Awan, Muhammad Ajmal Azad, et al. «A blockchain-inspired attribute-based zero-trust access control model for IoT». In: *Information* 14.2 (2023), p. 129.

[74] Catherine de Weever and Marios Andreou. «Zero trust network security model in containerized environments». In: *University of Amsterdam: Amsterdam, The Netherlands* (2020).

[75] John Bethencourt, Amit Sahai, and Brent Waters. «Ciphertext-policy attribute-based encryption». In: *2007 IEEE symposium on security and privacy (SP'07)*. IEEE. 2007, pp. 321–334.

[76] Nurmamat Helil, Kaysar Rahman, et al. «CP-ABE access control scheme for sensitive data set constraint with hidden access policy and constraint policy». In: *Security and Communication Networks* 2017 (2017).

[77] Vipul Goyal, Omkant Pandey, et al. «Attribute-based encryption for fine-grained access control of encrypted data». In: *Proceedings of the 13th ACM conference on Computer and communications security*. 2006, pp. 89–98.

[78] Jin Li, Xiaofeng Chen, et al. «Multi-authority fine-grained access control with accountability and its application in cloud». In: *Journal of Network and Computer Applications* 112 (2018), pp. 89–96.

[79] Sanjit Chatterjee and Palash Sarkar. *Identity-based encryption*. Springer Science & Business Media, 2011.

[80] Lan Zhou, Vijay Varadharajan, and Michael Hitchens. «Enforcing role-based access control for secure data storage in the cloud». In: *The Computer Journal* 54.10 (2011), pp. 1675–1687.

[81] Nazatul Haque Sultan, Vijay Varadharajan, et al. «A role-based encryption (rbe) scheme for securing outsourced cloud data in a multi-organization context». In: *IEEE Transactions on Services Computing* 16.3 (2022), pp. 1647–1661.

[82] Annapurna P Patil, Gaurav Karkal, et al. «Design and implementation of a consensus algorithm to build zero trust model». In: *2020 IEEE 17th India Council International Conference (INDICON)*. IEEE. 2020, pp. 1–5.

[83] Shahzaib Tahir, Sushmita Ruj, et al. «A new secure and lightweight searchable encryption scheme over encrypted cloud data». In: *IEEE Transactions on Emerging Topics in Computing* 7.4 (2017), pp. 530–544.

[84] Shaify Kansal and Meenakshi Mittal. «Performance evaluation of various symmetric encryption algorithms». In: *2014 international conference on parallel, distributed and grid computing*. IEEE. 2014, pp. 105–109.

[85] Tutubalin Pavel Innokentievich and Mokshin Vladimir Vasilevich. «The Evaluation of the cryptographic strength of asymmetric encryption algorithms». In: *2017 Second Russia and Pacific Conference on Computer Technology and Applications (RPC)*. IEEE. 2017, pp. 180–183.

[86] Hu Xiong, Yanan Zhao, et al. «Partially policy-hidden attribute-based broadcast encryption with secure delegation in edge computing». In: *Future Generation Computer Systems* 97 (2019), pp. 453–461.

[87] Jianghong Wei, Xiaofeng Chen, et al. «RS-HABE: Revocable-storage and hierarchical attribute-based access scheme for secure sharing of e-health records in public cloud». In: *IEEE Transactions on Dependable and Secure Computing* 18.5 (2019), pp. 2301–2315.

[88] Subhabrata Rana, Fatemeh Khoda Parast, et al. «A comprehensive survey of cryptography key management systems». In: *Journal of Information Security and Applications* 78 (2023), p. 103607.

[89] Ramaswamy Chandramouli, Michaela Iorga, and Santosh Chokhani. «Cryptographic key management issues and challenges in cloud services». In: *Secure Cloud Computing* (2013), pp. 1–30.

[90] Jin Li, Jingwei Li, et al. «Identity-based encryption with outsourced revocation in cloud computing». In: *IEEE Transactions on computers* 64.2 (2013), pp. 425–437.

[91] Jianghong Wei, Wenfen Liu, and Xuexian Hu. «Secure data sharing in cloud computing using revocable-storage identity-based encryption». In: *IEEE Transactions on Cloud Computing* 6.4 (2016), pp. 1136–1148.

[92] Shengmin Xu, Guomin Yang, and Yi Mu. «Revocable attribute-based encryption with decryption key exposure resistance and ciphertext delegation». In: *Information Sciences* 479 (2019), pp. 116–134.

[93] Muhammad Mujib and Riri Fitri Sari. «Performance evaluation of data center network with network micro-segmentation». In: *2020 12th International Conference on Information Technology and Electrical Engineering (ICITEE)*. IEEE. 2020, pp. 27–32.

[94] Nabeel Sheikh, Mayur Pawar, and Victor Lawrence. «Zero trust using network micro segmentation». In: *IEEE INFOCOM 2021-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE. 2021, pp. 1–6.

[95] Murshedul Arifeen, Andrei Petrovski, and Sergei Petrovski. «Automated microsegmentation for lateral movement prevention in industrial internet of things (iiot)». In: *2021 14th International Conference on Security of Information and Networks (SIN)*. Vol. 1. IEEE. 2021, pp. 1–6.

[96] Zirak Zaheer, Hyunseok Chang, et al. «eztrust: Network-independent zero-trust perimeterization for microservices». In: *Proceedings of the 2019 ACM Symposium on SDN Research*. 2019, pp. 49–61.

[97] Olli Mämmelä, Jouni Hiltunen, et al. «Towards micro-segmentation in 5G network security». In: *European Conference on Networks and Communications (EuCNC 2016) Workshop on Network Management, Quality of Service and Security for 5G Networks*. 2016.

[98] Amr Osman, Armin Wasicek, et al. «Transparent Microsegmentation in Smart Home {IoT} Networks». In: *3rd USENIX Workshop on Hot Topics in Edge Computing (HotEdge 20)*. 2020.

[99] Deepak Puthal, Saraju P Mohanty, et al. «Building security perimeters to protect network systems against cyber threats [future directions]». In: *IEEE Consumer Electronics Magazine* 6.4 (2017), pp. 24–27.

[100] Abdallah Moubayed, Ahmed Refaey, and Abdallah Shami. «Software-defined perimeter (sdp): State of the art secure solution for modern networks». In: *IEEE network* 33.5 (2019), pp. 226–233.

[101] Jaspreet Singh, Ahmed Refaey, and Juanita Koilpillai. «Adoption of the software-defined perimeter (sdp) architecture for infrastructure as a service». In: *Canadian Journal of Electrical and Computer Engineering* 43.4 (2020), pp. 357–363.

[102] Ahmed Sallam, Ahmed Refaey, and Abdallah Shami. «On the security of SDN: A completed secure and scalable framework using the software-defined perimeter». In: *IEEE access* 7 (2019), pp. 146577–146587.

[103] Iftekhar Ahmed, Tahmin Nahar, et al. «Protection of sensitive data in zero trust model». In: *Proceedings of the international conference on computing advancements*. 2020, pp. 1–5.

[104] Eslam Samy Hosney, Islam Tharwat Abdel Halim, and Ahmed H Yousef. «An artificial intelligence approach for deploying zero trust architecture (zta)». In: *2022 5th International Conference on Computing and Informatics (ICCI)*. IEEE. 2022, pp. 343–350.

[105] SM Mozammal Hossain, Russell Couturier, et al. «Automatic event categorizer for SIEM». In: *Proceedings of the 31st Annual International Conference on Computer Science and Software Engineering*. 2021, pp. 104–112.

[106] Christian Esposito. «Interoperable, dynamic and privacy-preserving access control for cloud data storage when integrating heterogeneous organizations». In: *Journal of Network and Computer Applications* 108 (2018), pp. 124–136.

[107] Ayan Sentuna, Abeer Alsadoon, et al. «A novel Enhanced Naive Bayes Posterior Probability (ENBPP) using machine learning: Cyber threat analysis». In: *Neural Processing Letters* 53 (2021), pp. 177–209.

[108] Paris Koloveas, Thanasis Chantzios, et al. «intime: A machine learning-based framework for gathering and leveraging web data to cyber-threat intelligence». In: *Electronics* 10.7 (2021), p. 818.

[109] Linn-Mari Kristiansen, Vinti Agarwal, et al. «Cti-twitter: Gathering cyber threat intelligence from twitter using integrated supervised and unsupervised learning». In: *2020 IEEE International Conference on Big Data (Big Data)*. IEEE. 2020, pp. 2299–2308.

[110] Andrea Tundis, Samuel Ruppert, and Max Mühlhäuser. «On the automated assessment of open-source cyber threat intelligence sources». In: *Computational Science–ICCS 2020: 20th International Conference, Amsterdam, The Netherlands, June 3–5, 2020, Proceedings, Part II 20*. Springer. 2020, pp. 453–467.

[111] Peng Gao, Xiaoyuan Liu, et al. «A system for automated open-source threat intelligence gathering and management». In: *Proceedings of the 2021 International conference on management of data*. 2021, pp. 2716–2720.

[112] Giuseppe Cascavilla, Damian A Tamburri, and Willem-Jan Van Den Heuvel. «Cybercrime threat intelligence: A systematic multi-vocal literature review». In: *Computers & Security* 105 (2021), p. 102258.

[113] Thomas D Wagner, Khaled Mahbub, et al. «Cyber threat intelligence sharing: Survey and research directions». In: *Computers & Security* 87 (2019), p. 101589.

[114] Xuren Wang, Jie Yang, et al. «Threat Intelligence Relationship Extraction Based on Distant Supervision and Reinforcement Learning.» In: *SEKE*. 2020, pp. 572–576.

[115] Mohit Sewak, Sanjay K Sahay, and Hemant Rathore. «Deep reinforcement learning for cybersecurity threat detection and protection: A review». In: *International Conference On Secure Knowledge Management In Artificial Intelligence Era*. Springer. 2021, pp. 51–72.

[116] Min Du, Feifei Li, et al. «Deeplog: Anomaly detection and diagnosis from system logs through deep learning». In: *Proceedings of the 2017 ACM SIGSAC conference on computer and communications security*. 2017, pp. 1285–1298.

[117] Yumin Wang and Zixiang Ji. «Design and implementation of a semi-supervised anomaly log detection model dda». In: *2021 International Conference on Computer Communication and Artificial Intelligence (CCAI)*. IEEE. 2021, pp. 86–90.

[118] Jacopo Soldani and Antonio Brogi. «Anomaly detection and failure root cause analysis in (micro) service-based cloud applications: A survey». In: *ACM Computing Surveys (CSUR)* 55.3 (2022), pp. 1–39.

[119] Max Landauer, Florian Skopik, et al. «System log clustering approaches for cyber security applications: A survey». In: *Computers & Security* 92 (2020), p. 101739.

[120] Raghavendra Chalapathy and Sanjay Chawla. «Deep learning for anomaly detection: A survey». In: *arXiv preprint arXiv:1901.03407* (2019).

[121] Gamal Eldin I Selim, EZZ El-Din Hemdan, et al. «Anomaly events classification and detection system in critical industrial internet of things infrastructure using machine learning algorithms». In: *Multimedia Tools and Applications* 80.8 (2021), pp. 12619–12640.

[122] Xiaohan Sun, Guanghong Yang, and Jilan Zhang. «A real-time detection scheme of user behavior anomaly for management information system». In: *2020 IEEE 4th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*. Vol. 1. IEEE. 2020, pp. 1054–1058.

[123] Yanping Tang, Bingxian Ma, and Zheng Wu. «Research on user clustering algorithm based on software system user behavior trajectory». In: *Proceedings of the 2nd International Conference on Big Data Technologies*. 2019, pp. 11–14.

[124] Wawan Yunanto and Hsing-Kuo Pao. «User Behaviour Risk Evaluation in Zero Trust Architecture Environment». In: *2022 IEEE 8th World Forum on Internet of Things (WF-IoT)*. IEEE. 2022, pp. 1–6.

[125] G Ramesh, J Logeshwaran, and V Aravindarajan. «A secured database monitoring method to improve data backup and recovery operations in cloud computing». In: *BOHR International Journal of Computer Science* 2.1 (2022), pp. 1–7.

[126] Eva Weishäupl, Emrah Yasasin, and Guido Schryen. «Information security investments: An exploratory multiple case study on decision-making, evaluation and learning». In: *Computers & Security* 77 (2018), pp. 807–823.

[127] Belal Ali, Simsam Hijjawi, et al. «A maturity framework for zero-trust security in multiaccess edge computing». In: *Security and Communication Networks* 2022 (2022).

[128] William Yeoh, Marina Liu, et al. «Zero trust cybersecurity: Critical success factors and A maturity assessment framework». In: *Computers & Security* 133 (2023), p. 103412.

[129] Zillah Adahman, Asad Waqar Malik, and Zahid Anwar. «An analysis of zero-trust architecture and its cost-effectiveness for organizational security». In: *Computers & Security* 122 (2022), p. 102911.

[130] Elisa Bertino. «Zero trust architecture: does it help?» In: *IEEE Security & Privacy* 19.05 (2021), pp. 95–96.

[131] Cătălin Adrian Iordache, Adrian Viorel Dragomir, and Constantin Viorel Marian. «Public institutions updated enhanced biometric security, zero trust architecture and multi-factor authentification». In: *2022 International Symposium on Electronics and Telecommunications (ISETC)*. IEEE. 2022, pp. 1–4.

[132] Justice Opara-Martins, Reza Sahandi, and Feng Tian. «Critical analysis of vendor lock-in and its impact on cloud computing migration: a business perspective». In: *Journal of Cloud Computing* 5 (2016), pp. 1–18.

[133] Taisho Sasada, Masataka Kawai, et al. «Factor Analysis of Learning Motivation Difference on Cybersecurity Training with Zero Trust Architecture». In: *IEEE Access* (2023).

[134] Umer Ahmed Butt, Rashid Amin, et al. «Cloud security threats and solutions: A survey». In: *Wireless Personal Communications* 128.1 (2023), pp. 387–413.

[135] Dhaivat Dave, Nayana Meruliya, et al. «Cloud security issues and challenges». In: *Big Data Analytics: Proceedings of CSI 2015*. Springer. 2018, pp. 499–514.

[136] I Indu, PM Rubesh Anand, and Vidhyacharan Bhaskar. «Identity and access management in cloud environment: Mechanisms and challenges». In: *Engineering science and technology, an international journal* 21.4 (2018), pp. 574–588.

[137] Hussain AlJahdali, Abdulaziz Albatli, et al. «Multi-tenancy in cloud computing». In: *2014 IEEE 8th international symposium on service oriented system engineering*. IEEE. 2014, pp. 344–351.

[138] Ashish Singh and Kakali Chatterjee. «Cloud security issues and challenges: A survey». In: *Journal of Network and Computer Applications* 79 (2017), pp. 88–115.

[139] Harsh Gupta and Deepak Kumar. «Security threats in cloud computing». In: *2019 International Conference on Intelligent Computing and Control Systems (ICCS)*. IEEE. 2019, pp. 1158–1162.

[140] Shengmei Luo, Zhaoji Lin, et al. «Virtualization security for cloud computing service». In: *2011 International Conference on Cloud and Service Computing*. IEEE. 2011, pp. 174–179.

[141] Guodong Zhu, Yue Yin, et al. «Detecting virtualization specific vulnerabilities in cloud computing environment». In: *2017 IEEE 10th International Conference on Cloud Computing (CLOUD)*. IEEE. 2017, pp. 743–748.

[142] Antony Martin, Simone Raponi, et al. «Docker ecosystem–vulnerability analysis». In: *Computer Communications* 122 (2018), pp. 30–43.

[143] Gaurav Somani, Manoj Singh Gaur, et al. «DDoS attacks in cloud computing: Issues, taxonomy, and future directions». In: *Computer communications* 107 (2017), pp. 30–48.

[144] Anat Bremler-Barr, Eli Brosh, and Mor Sides. «DDoS attack on cloud auto-scaling mechanisms». In: *IEEE INFOCOM 2017-IEEE Conference on Computer Communications*. IEEE. 2017, pp. 1–9.

[145] Meraj Mostam Kashi, Anis Yazidi, and Hårek Haugerud. «Mitigating Yo-Yo attacks on cloud auto-scaling». In: *2022 14th IFIP Wireless and Mobile Networking Conference (WMNC)*. IEEE. 2022, pp. 46–53.

[146] Anat Bremler-Barr and Michael Czeizler. «Tandem Attack: DDoS Attack on Microservices Auto-scaling Mechanisms». In: *IEEE INFOCOM 2023-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE. 2023, pp. 1–2.

[147] Raihan ur Rasool, Hua Wang, et al. «A survey of link flooding attacks in software defined network ecosystems». In: *Journal of Network and Computer Applications* 172 (2020), p. 102803.

[148] Mauro Conti, Nicola Dragoni, and Viktor Lesyk. «A survey of man in the middle attacks». In: *IEEE communications surveys & tutorials* 18.3 (2016), pp. 2027–2051.

[149] Mohammed N Alenezi, Haneen Alabdulrazzaq, et al. «Evolution of malware threats and techniques: A review». In: *International journal of communication networks and information security* 12.3 (2020), pp. 326–337.

[150] Mehrdad Jangjou and Mohammad Karim Sohrabi. «A comprehensive survey on security challenges in different network layers in cloud computing». In: *Archives of Computational Methods in Engineering* 29.6 (2022), pp. 3587–3608.

[151] Jose Moura and David Hutchison. «Review and analysis of networking challenges in cloud computing». In: *Journal of Network and Computer Applications* 60 (2016), pp. 113–129.

[152] Subashini Subashini and Veeraruna Kavitha. «A survey on security issues in service delivery models of cloud computing». In: *Journal of network and computer applications* 34.1 (2011), pp. 1–11.

[153] OWASP API Security. URL: https://owasp.org/www-project-api-security/ (last visited: Apr. 2, 2024).

[154] Hassan Jamil Syed, Abdullah Gani, et al. «Cloud monitoring: A review, taxonomy, and open research issues». In: *Journal of Network and Computer Applications* 98 (2017), pp. 11–26.

[155] Microsoft. URL: https://www.microsoft.com/en-us/security/business/zero-trust (last visited: June 5, 2024).

[156] VMWare. URL: https://www.vmware.com/solutions/zero-trust-security.html (last visited: June 5, 2024).

[157] Cisco. URL: https://www.cisco.com/site/us/en/solutions/security/zero-trust/index.html (last visited: June 5, 2024).

[158] Zscaler. URL: https://www.zscaler.com/platform/zero-trust-exchange (last visited: June 5, 2024).

[159] Cloudflare. URL: https://developers.cloudflare.com/cloudflare-one/ (last visited: June 5, 2024).

[160] Akamai. URL: https://www.akamai.com/solutions/security/zero-trust-security (last visited: June 5, 2024).

[161] Fortinet. URL: https://www.fortinet.com/solutions/enterprise-midsize-business/network-access (last visited: June 5, 2024).

[162] CrowdStrike. URL: https://www.crowdstrike.com/solutions/zero-trust/ (last visited: June 5, 2024).

[163] Tailscale Docs website. URL: https://tailscale.com/kb/1017/install (last visited: June 5, 2024).

[164] Tailscale. URL: https://tailscale.com/ (last visited: June 5, 2024).