JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

Zero Trust Architecture in the Integrated Cloud, Edge and IoT environment

[1]Kailash, [2]Saswati Mukherjee

[1]Centre for Development of Advanced Computing (C-DAC), Chennai, India
[2]College of Engineering, Guindy, Anna University, Chennai, India

Abstract : Traditional Cloud services may be unable to meet the computational service needs of many kinds of Service Consumers, given the fluctuating demand for these services. Despite the emergence of Multi Clouds and Cross-Clouds, there exists a substantial demand for Fog and Edge systems to address the requirements of IoT. It is clear that the current necessity is an environment that presents all frameworks and options to both Consumers and Providers according to the demands and needs of the present task. This necessitates a harmonious amalgamation of Cloud with Fog and Edge environments capable of interfacing with IoT devices via IoT gateways situated in proximity to the devices. Contemporary progress in computing, storage, and networking demands the creation of innovative security frameworks to protect the Cloud environment amid its intricacies. The study suggests the implementation of a Security Manager to protect the Integrated ecosystem and facilitate uninterrupted operations. The suggested Security Manager emphasizes Identity and Access Management by executing Authentication and Authorization with Zero Trust principles. Given the dynamic nature of the environment, characterized by diverse Providers and Consumers, it is essential to protect resources and services from the ground up. The Security Manager executes Security Orchestration utilizing Micro segmentation alongside suitable algorithms.

IndexTerms **– Zero Trust Architecture, Integrated Cloud.**

I. INTRODUCTION

Diverse hardware and software services linked to compute and storage are needed by emerging technologies. No single entity, such as public or private clouds or edge environments, can provide all of the service consumer needs. Nonetheless, these settings can be integrated with service arbitration to meet workload demands and offer customers smooth service. To successfully handle the obstacles and problems, an integrated environment made up of Edge and fog nodes, public and private clouds, hybrid clouds, IoT devices, and gateways is required. In order to safeguard the Cloud ecosystem given its complexity, new security models must be developed in light of recent developments in processing, storage, and network technology. Additional attack surfaces have been created by the greater decentralization of networks brought about by the migration of transactions to the cloud. The intrinsic heterogeneity of the Integrated Environment makes it complex.

For smooth operations in such a setting, zero trust concepts are quite effective (2). ZTA makes the assumption that no entity—internal or external to the system—is inherently unreliable, including Edge nodes, IoT gateways, cloud services, and resources like networking, hardware, and software. Security issues, such as a lack of confidence and insufficient validation of resources and procedures, are brought about by the diversity of services and platforms in an integrated environment. A potentially useful answer to the growing security problems present in this constantly changing ecosystem can be found by integrating ZTA into the Integrated Environment. A zero trust framework can be used to counteract risks including insider attacks, denial of service attacks, and misconfigured gateways, even while many security measures are in place to safeguard the perimeter of clouds, edge nodes, and IoT gateways. Complete openness about all system resources, users, and data flows is necessary to achieve Zero Trust in the Integrated Environment. Workloads are divided into distinct, isolated zones or segments using the microsegmentation security technique, and policies govern the movement and access of data and applications within those zones. Each segment's workloads or services will have workload-specific security limitations, dynamically configurable policies, and least-privileged access. By putting controls in place that limit traffic into and out of designated workloads, the objective is to reduce threat mobility. Administrators can control security policies that restrict traffic according to the least privilege and zero trust principles by using microsegmentation. Using automatic software-defined techniques to divide traffic across all entities in the Integrated Environment is known as microsegmentation. The idea behind security automation is to use automation to lessen the frequency with which security experts must intervene. In the dynamic integrated environment, security threats are constantly evolving. In order to manage security in this complex context, Security Orchestration and Automation (SOAR) elements (3) are crucial.

By concentrating on Identity and Access Management (IdAM) and microsegmentation to guaranty the smooth operation of the Integrated environment, this paper proposes to effectively mitigate some of the critical threats (communication and service access, infrastructure, and network) as mentioned in Figure 1. The bulk of security vulnerabilities are mitigated by implementing the Zero Trust Architecture in the Integrated environment via the Security Manager, as suggested by this research. The proposed Security Manager includes two major components:

i. IdAM that includes authentication, authorization and verification
ii. Security Orchestration with Microsegmentation.

While the proposed Security Manager takes care of the majority of critical components of the Zero Trust Architecture, one of the other critical components, viz., Policy and trust engine is not considered in the scope of this paper.

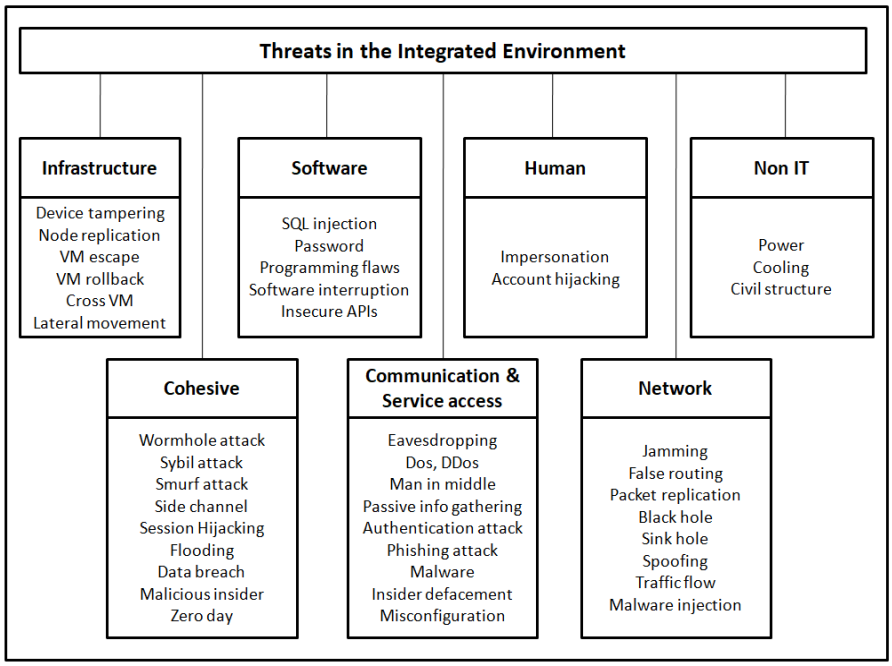

Figure 1 Categorization of Security threats in Integrated environment.

Access control administration is a challenging endeavor that is fraught with a variety of issues due to the dynamic, diversified, and integrated environment. Security is the fundamental criterion for identity and access management in an integrated environment, regardless of whether it is centralized or distributed, and includes technical compatibility and interoperable architecture. Different Service Providers and Consumers are unable to integrate or be interoperable due to the complexity. In order to guaranty uninterrupted operations, the conventional authentication methods of Service Providers necessitate an additional layer of protection. Software Defined Networking (SDN) and Network Function Virtualization (NFV) provide a multitude of advantages for optimizing and automating security management through Virtual Network Functions (VNF)s by enhancing end-to-end security in the integrated environment. The Virtual Network Functions are essentially the SDN function implementations. NFV is a network architecture concept that aims to virtualize the VNFs that include network functions like firewalls, routers, load balancers, intrusion detection systems and so on. VNFs allow these network functions to run as software on standard, off-the-shelf servers. VNF clustering refers to the practice of grouping multiple instances of the same Virtual Network Function together to operate as a single, logical entity, to enhance network reliability, performance, and scalability. VNF clustering leverages virtualization to create highly resilient and scalable network functions. VNF clustering is crucial for ensuring high availability and performance in integrated environment.

II. LITERATURE SURVEY

Chuanyi Liu et al. (4) created a framework for encrypted data search and sharing based on the Cloud Access Security Broker (CASB). They showed how CASB can be used to keep applications working and provide strong security in cloud environments. They also talked about the architecture of the CASB framework, which includes managing keys, creating search indexes, encrypting and decrypting data, and running search queries. In spite of this, the study found that most CASB solutions offered by for-profit companies lack comprehensive knowledge about important strategies and details. This makes it hard for beginners to set the stage for future research. Annapurna P. Patil et al. (5) came up with a hybrid way for nodes to agree on something based on a shared value. It's a mix of a changed proof of work and a changed proof of time that has passed. Nodes can set up a Zero Trust paradigm through this process of building agreement. In 2021, Truong Thu Huong et al. created the LocKEdge security system design for Internet of Things networks, with a focus on Edge layer low-complexity cyberattack detection. The design combines federated and centralized learning modes to get the most out of the detection method. In the feature extraction module, Principal Component Analysis is suggested for reducing the number of dimensions. In the multi-attack recognition module, a neural network is suggested for classifying the features. The literature study doesn't talk about other important topics that should have been covered, like anomaly detection, intrusion detection systems, and machine learning techniques that are often used in IoT security. Instead, it mainly talks about ways to find threats, federated learning, Edge computing, and SDN-based frameworks. Xie et al. (6) describe a ZTA-based micro-segmentation network security method. The security gateway module handles authentication and permission for north-south traffic and adaptive control for traffic going east to west. It can't, however, provide in-depth analysis or complete solutions for problems that have been discovered. To get around these problems and give the business more useful information, more research is needed. Mengyu Ma et al. (7) suggested Hybrid Dynamic-static Analysis algorithms, which use both dynamic and static analysis algorithms to find valid access relationships between services. These are then used to build network segments. The Automated Policy Engine introduces a policy translator called APE, which takes the results of the analysis algorithms and turns them into low-level policies so that deployments can be done automatically. The writers didn't talk about the problems or limits of current micro-segmentation solutions. They also didn't do a full analysis of how well clustering algorithms work for finding segments in a network that behave consistently. This kind of analysis could tell us a lot about how well different

segmentation methods work. Xiao et al. (8) used an ant colony-based approach to create a cross-security domain dynamic orchestration algorithm. The algorithm creates Service Function Chaining for network security functions, optimizing for load balancing, shortest path, and minimum delay. It was made for industrial vertical scenarios. The deployment problem of the dynamic orchestration method was fixed by this algorithm. Bringhenti et al. (9), suggest setting up the security of multi-cluster and multi-domain Kubernetes environments with a top-level entity called Multi-Cluster Orchestrator. This entity sets network security policies automatically for each cluster in the environment. The goals of this method are to make cross-cluster links more clear and to cut down on configuration mistakes. In their paper (10) Vinoth et al. suggested anonymous pre- and post-authentication, a useful Cloud-based mutual key agreement system in which users' anonymous authentication is done by the Cloud server creating a fake name for them. This is a mutual key agreement method that makes it easier to talk to each other and uses little computer power. It cuts down on the time and bits needed to send data between devices, users, and Cloud resources. This lowers the cost of storage while data is being sent from the Cloud server to the user. On the other hand, it doesn't look at possible limits, security holes, or problems with scaling. In his 11th paper, Xiaotong Zhou suggests a framework for anonymous and auditable distributed access control in edge computing. This framework includes conditional anonymous identification and auditable multiple attribute-based encryption. The paper also includes formal security models, security proofs, a qualitative comparison, and a speed benchmark. However, it doesn't go into detail about how these things can be used in specific Edge computing scenarios. Miloud Bagaa et al. (12) showed a low-cost, highly effective system for managing different Service Function Chains in virtualized and software-defined IoT networks. They came up with a cost-effective orchestration system that takes into account Quality of Service (QoS), network security levels, and the resource capabilities of Virtual Network Function Instances. It showed a framework for the cost-effective management of network slicing, lifecycle management of SFCs, and zero-touch network service management of different network slices in a distributed IoT environment using SDN and NFV technologies. It also had a heuristic solution for effective lifecycle management of SFCs and an optimal solution using linear integer programming. But it's not clear how the suggested system will work in terms of scalability and performance. The work of Phu Nguyen et al. (13), shows ongoing work on a framework that has multiple SOAR decision-making and orchestration tools, using adaptation intelligence based on Reinforcement Learning, virtual reality, avatar-human interaction, and advanced Cyber Threat Intelligence tools. Aytaj Badirova et al. (14) give basic information on the topic and group the problems of managing access in both centralized and decentralized identity governance. Traditional methods like federated and controlled identity have been looked into, as well as more modern ones like blockchain-based decentralized identity, AI/ML access management, and ABE schema.

III. SECURITY MANAGER ARCHITECTURE

To provide total safety and protection of data, this component employs a zero-trust architecture, limits access, identifies threats, manages data security, and provides granular access control. Strong authentication, authorization, and encryption are all parts of the Zero-trust security architecture that the security manager strictly enforces. The Security Manager incorporates two planes into its distributed architecture namely,

- Access plane
- Management plane

The management plane components are likely to act as server and the access plane components are likely to act as clients, in the security perspective.

3.1 Key functionalities proposed on the planes

Although there are various security controls to be employed, to secure the integrated environment against the threats mentioned above, the chapter proposes the following major functionalities, on the management and access planes, since enforcing them will mitigate the threat categorized above including infrastructure, network, software, human, communication and service access.

- Identity and access management,
- Microsegmentaiton with security orchestration,
- Security Gateway Service

The identity and access management and microsegmentation functionalities are positioned in the management plane as these functionalities are controlled in centralized manner by the security manager whereas the entry points and security enforcement points, distributed across the Integrated environment are positioned in the access plane as they are the source as well as target (receipt) of various security controls to be enforced by the management plane. The entry points and security points referred in the access plane belongs either the service provider or service consumer roles. The IdAM and microsegmentation components in management plane, entry and security enforcement points in the access plane are depicted in Figure 2.

IdAM involves discovery, identity proofing, identity aggregation, identity deactivation, whereas the microsegmentation involves the SDN orchestrator and VNF instances.

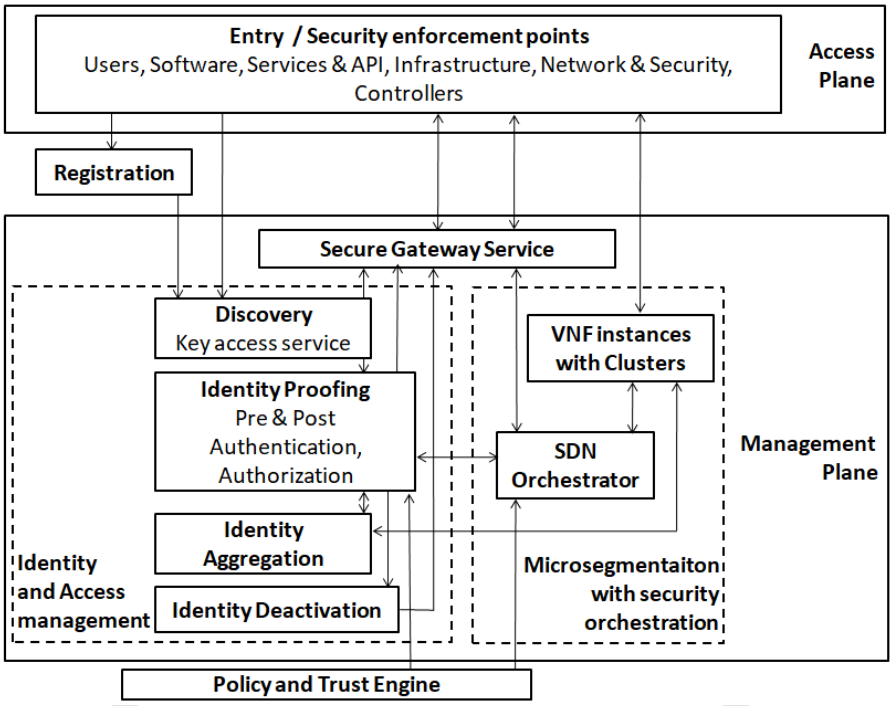

Figure 2 Security Manager Components

3.1.1 Registration module

All the entities, either service provider or service consumer, positioned in access plane, participating in the integrated environment should register with the Global manager as explained in chapter 3 and chapter 4 of the paper. Upon registration, every entity among the entry / security enforcement points, initiates access request to the discovery module.

3.1.2 Policy and Trust Engine

The policy and trust engine holds dynamic policies with access control for the services and resources in the Integrated environment. The identity proofing under IdAM and SDN orchestrator under microsegmentation adopts the policy enforcement from this policy and trust engine to ensure security of the Integrated environment.

3.1.3 Access plane

The access plane encompasses all services and resources, essentially a variety of entrance and security enforcement points, distributed throughout the Integrated Environment. In alignment with instructions from the management layer, security protocols are implemented in the access layer for each access request. All the components, that are part of the Integrated environment, whether from the service provider end or from the service consumer end are positioned under access plane, since Zero trust implementation mandates each and every access request from every component to be validated to ensure seamless secure operations. The components of access plane are categorized as follows,

- Users - Service Consumers, Cloud administrators, solution providers, developers and support personnel.
- Software - Platform, Database, Operating System and all other software used in the integrated environment.
- Services & API - Application/Software service, APIs of corresponding software and services.
- Infrastructure Devices - Cloud servers, storage, non IT equipment (Power and cooling, BMS components), Edge nodes, IoT gateways.
- Network & Security - Network equipments (e.g., switches, routers, load balancers), Security devices (e.g., firewall, WAF, packet shapers), physical connectivity links.
- Controllers – Virtualization, Containers, Cloud orchestration, Management and controlling software.

The above-mentioned components in the access plane, as source of workflow, initiates an security access request, which are processed by the management plane components, and upon successful verification and validation, the access is granted as part of IdAM. The microsegmentation is enforced in every access request's corresponding destination components of the access plane under the directive of Management planes's security orchestration.

3.1.4 Management Plane

This serves as the Security Manager's brain and is a crucial part of the system. The management plane's components are in charge of the Integrated environment's security control. The management plane issues all necessary security directives, including those pertaining to granting access, handling security-related requests, and carrying out security orchestration-related tasks. Identity and access management as well as security orchestration components are found in the management plane.

3.1.5 Secure Gateway Service

Security Gateway Service (SGS) acts as an intermediate communication point between the IdAM, microsegmentation and entry, security enforcement points. Since the integrated environment consists of heterogeneous components including Clouds, Edge nodes in the access plane, communication of these access plane components with the management plane components possess security risks. Hence SGS is introduced to ensure seamless security workflow for authentication, authorization, verification, security orchestration and microsegmentation of access plane components.

3.1.6 Identity and Access Management

The identity and access management capability guarantees that all entities in the Integrated Environment, from Global Manager to last mile services and devices in the access plane, are authenticated, authorized, verified, and validated during workflow execution. Although the entities in the access plane may be dependable in a traditional environment, the Zero Trust

architecture requires the verification and validation of the identities of all entities at all instances, and is regulated throughout the execution period. The IdAM process involves the following four modules,

- Discovery
- Identity proofing
- Identity aggregation
- Identity deactivation

Discovery

The discovery module, as part of the security process, continuously monitors all the entities that hold an active session and requested for access within the integrated environment, regardless of their trustworthiness or position (internal or external). Any delta changes in the entity's workflows are recognized by the discovery process and enforced to the verification and validation in accordance to the Zero Trust implementation. The delta changes are monitored by the analysis of the header information, signature stamp and meta data information of entities. The discovery process interacts with the entry / security enforcement points in the access plane, registration module, SGS, identity proofing module.

Identity proofing

Identity Proofing is the process of verifying and validating an entity's identity. It is not merely a matter of verifying an individual's password and identification. In a Zero Trust system, identity validation is a critical and ongoing process. It establishes the foundation for secure service access in a controlled environment by verifying the legitimacy of entities and services in the Integrated Environment.

- **Authentication request & authentication code**
  In the proposed system, the entity sends an authentication request to SGS. To gain access from the Security Manager, the entity needs to prove its authenticity using the authentication code that the SGS sent to it. SGS verifies the authentication request and delivers authentication code to Identity proofing. This authentication code is validated by Identity proofing based on the Policy and trust engine rules, and result passed on to SGS. Upon validation, a pseudo-identity code is generated by SGS.
- **Pseudo-identity**
  Pre and post-authentication are the crucial process of Identity proofing. As part of the process, the pseudo-identity is generated by the SGS service for every session, using the entity data, and the same is shared to the entity for every session. The identity proofing process verifies the authenticity of incoming request from entity using the pseudo-identity. This pseudo-identity is used for mutual authentication between the entity and SGS.
- **Session keys**
  Upon mutual authentication, SGS distributes the session key to entity and the services, for encryption. The entity (source of service request) and the services (destination of service request) exchange their corresponding session keys, thereby initiating the workload access.
- **Key management**
  The service provider makes the public and private keys and sets the rules for entry. There is a header in every piece of data and every access request. One part of the entity name is the access tree, which is based on information from the policy and trust engine. If the encrypted data matches the header information in the access tree, the data can only be decrypted. This is based on the policy and trust engine. The access tree is a part of the entity key that makes the characteristics different. You can use the access tree to decrypt data only if it matches the properties of the tree.

By accomplishing pre and post-authentication, the suggested method entirely safeguards entity privacy. The suggested system's Security Manager creates the identity using entity data. The entity has no idea that this false identity is being utilized to verify the requester's legitimacy.

The Workflow of the identity proofing is depicted in the figure 3.

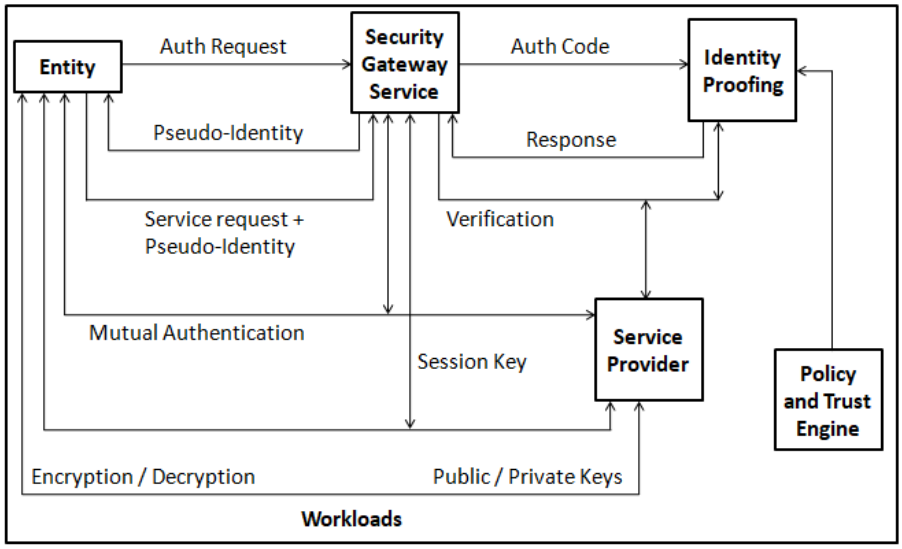

Figure 3 Identity Proofing

Identity aggregation

The Security Manager collects identity information from various sources, including Identity Proofing Points like digital certificate authorities, Multi-Factor Authentication servers, firewalls, access control lists, and identity providers, as well as Security

Enforcement Points such as firewalls, access control lists, and endpoint security agents. Identity consolidation via SGS amalgamates, integrates, and assesses the identity information and access control profiles of each entity for every session, as it traverses the Integrated Environment from various security enforcement nodes. Throughout each session, users and devices may encounter numerous security enforcement points within a dynamic, cohesive environment comprising Cloud, Edge, and IoT gateways. A fundamental component of the Security Manager framework, identity aggregation, employs many strategies to ensure secure and efficient access management. • Consolidate Identity Profiles: Utilizing the access tree characteristics, the aggregated information is synthesized to construct a detailed identity profile for each entity. • Analyzing profiles for access regulation: In accordance with established policies from the Policy and Trust Engine, the Security Manager assesses the amalgamated identity profiles to ascertain the suitable access rights for each entity.

When entities are administered collectively for a specific session, rather than individually, the strain on access control is alleviated. While alternative techniques exist, group keys are a prevalent approach for managing groups. Securing the information by encryption and providing individuals with the key is a simple remedy. Given that entity groups are fluid, the management system must possess sufficient flexibility to adapt to security modifications anytime an entity is introduced or eliminated. Numerous group key schemas exist, including triple-party key agreement protocols, as well as static and dynamic group key schemas, contingent upon the implemented system and service as delineated by Yilong Zheng et al. (1). The architecture of the Security Manager employs a strong and centralized methodology for access management in the Integrated Environment through the use of identity aggregation. Besides improving user experience and ensuring accurate access determinations based on real-time identification data, it bolsters the security of the linked ecosystem.

Identity deactivation

Upon completion of workload execution for every session, one of the crucial processes is cleansing of the credentials and session keys. In the proposed approach, the pseudo-identity, session key and the access tree are to be deactivated, as per the Zero trust implementation principle, to ensure every workflow is to be verified and validated by default. Identity deactivation ensures a safe exit of workloads from the Integrated environment. The Identity deactivation is considered as a future work.

IV. WORKFLOW

The workflow of Identity and Access Management within the Integrated environment follows,

4.1 Authentication request initiation and discovery function

A random binary integer is accepted as input by the Security Manager, which then selects a one-way hash function $h:\{0,1\}^* \in Z_q^*$ with a fixed-size output. The output hash result can be significantly affected by even minor modifications to the input string, as previously mentioned. Both entities and the SGS service have access to this h function.

4.1.1 Pseudo-identity

The entity finding feature computes a pseudo-identity value (PU_i) for every entity to safeguard their anonymity. SGS generates two large prime numbers, p and q, ensuring that q < p - 1. It subsequently selects g as the generator of q. It then computes the master key $k \in Z_q^*$. This master key owned by SGS is used to generate the session key.

4.1.2 Mutual authentication and session keys

The SGS provides the entity with a session id, UID_i once the entity successfully authenticates in using its pseudo-identity. Subsequently, the entity computes Z and K, using a random element $d \in Z_q^*$.

$$Z_i = h(PW_i \oplus d, UN_i, UID_i)$$
$$K_i = h(PWi \oplus d, UN_i, Y_i)$$

The entity transmits $\{UID_i, Z_i \oplus d, PW_i \oplus d\}$ to the SGS for authentication. The Authentication module uses this to retrieve the values of Z_i and d from $Z_i \oplus d,\ PW_i \oplus d$, respectively. Next, in the following manner, the Identity proofing calculates the entity's pseudo-identity (PU_i) as $PU_i = E_k(Y_i \oplus UID_i)$.

The SGS then executes an authentication code, $AC_u = \{l_3,\ l_4, Y_i\}$ and sends it to the user as $Z_i \oplus l_3, Z_i \oplus l_4$, and calculates,

$$l_1 = h(PU_i, Y_i)$$

$$l_2 = l_1 \parallel C$$

$$l_3 = l_2 \oplus Z_i$$

$$l_4 = PU_i \oplus Z_i \oplus h(Y_i)$$

Upon getting this, the entity extracts the values l3 and l4 and reconstructs $AC_u = \{l_3, l_4, Y_i\}$. The entity uses this ACu credential when the authentication module verifies their legitimacy.

To compute the session key, the Gateway service selects random number $a_2 \in Z_q^*$ to compute

$$S_k = h(\upsilon_1, Y_i, PU_i)$$

$$\upsilon_\tau = S_k \oplus \upsilon_1$$

The authentication module receives $(v_7, a_2 \oplus v_1)$ from the Gateway service and formulates $S'_k = h(v'_1, Y_i,$ and verifies $v'_7 = S'_k \oplus v'_1$.

Upon successful verification S'_k is accepted, else rejected.

When accessing the services, Gateway service uses a random number $a \in Z_q^*$ and computes

$$v_1 = C^a \bmod p$$

$$v_2 = l_4 \oplus Z_i$$

$$v_3 = v_1 \oplus v_2$$

$$v_4 = g^a \bmod p \quad v_5 = h(v_1, v_2, v_4, T_i) \quad v_6 = v_5 \oplus l_1$$

where Ti represents the current time stamp.

The authentication module receives a message from the Gateway service. After obtaining these variables, the authentication module calculates $v'_1 = v_4^b \bmod p$

$$v'_2 = v'_1 \oplus v_3$$

$$v'_5 = h(v'_2, v_4, v'_1, T_i)$$

$$l'_1 = h(PU_i, Y_i)$$

$$v'_6 = l'_1 \oplus v'_5$$

If, $v'_6 = v_6$, then the authentication module authorize the Gateway service.

To compute the session key (S_u) between the entities and Integrated environment components, the authentication module again chooses a random number $a_3 \in Z_q^*$ and formulates,

$$S_u = h(PW_i, l_4 \oplus a_3, C)$$

$$v_8 = S_u \oplus l_3$$

The authentication module returns $(v_8, a_3 \oplus l_3)$ to the entities.

Upon receipt, the authentication module calculates, $S'_u = h(PW_i, l_4 \oplus a_3, C)$ and verifies, $v'_8 = S'_u \oplus l_3$. If validated upon verification, S'_u is accepted by the entities, else rejected.

4.1.3 Public / private key

The service provider computes its public key, $C = g^b \bmod p$, using a random number, $b \in Z_q^*$, as its private key. In a similar vein, the entities generates its public key, $Y_i = g^{X_i} \bmod p$, using a random number $X_i \in Z_q^*$ as its private key $E_k(.)$ is the encryption function that utilizes the master key k. The Authentication module sends this PU_i to the Gateway service, together with $\{PU_i, Z_i\}$, via a secure channel.

4.2 Microsegmentation

Microsegmentation and Virtual Network Functions (VNFs) work in collaboration to improve security and adaptability in the integrated environment. Microsegmentation enables specific control over interactions between different workloads, allowing each workload to have its own customized security policies, as per Zero Trust Principles. The Zero Trust security mechanism is implemented in the integrated environment by using VNFs to deliver security functionalities within microsegmented services. By orchestrating security functions with VNFs, manual effort required to manage security policies can be reduced, allowing more efficient use of resources and reduces the risk of human error and threats. The integration of Virtual Network Functions with security orchestration is crucial for creating dynamic and responsive security architectures. Security orchestration platforms deploy microsegmented services on VNFs based on predefined policies and trust engine in response to security events. The security orchestration allows for the dynamic application of security policies to VNFs and enables the creation of micorsegmented services where traffic is routed through a sequence of VNFs to apply multiple security functions. VNFs provide the security tools, whereas security orchestration provides the automation to deploy, configure, and manage those tools in an intelligent way. This combination creates a much more responsive and robust security system. The approach focuses on dividing a workflow segments into very small, isolated zones, thereby limiting the circumference of a security breach, preventing attackers from moving laterally. The security orchestration tasks include automation of the management of security tools and workflows, including tasks like policy enforcement, threat response, and compliance reporting and so on. Security orchestration platforms can automate the creation and enforcement of microsegmentation policies.

The access plane components depend majorly on VNFs for network functionalities. Virtual network functions are used to provide conventional network features including intrusion detection systems, load balancers, firewalls, and more. VNFs can be used to split the network into smaller areas with different levels of security. This limits an attacker's ability to move laterally after gaining initial access to a certain area. The SDN Orchestrator, an essential component of the Zero Trust framework, serves a crucial role in managing VNFs inside the integrated environment by automating VNF provisioning, configuration, and lifecycle management throughout their deployment.

The VNF clustering is implemented through active-active or active-standby clustering models, depending on latency and failover requirements in Integrated environment. As demands increases, more VNF instances can be scaled within the cluster, distributing the load and preventing bottlenecks, particularly important for Edge nodes and IoT gateways, where traffic patterns can be highly variable. Implementing strong authentication and authorization mechanisms to prevent unauthorized access to the cluster through the proposed IdAM is one of the crucial functions in the Integrated environment. VNF clusters may need to perform protocol translation and security filtering.

4.3 Process

As and when a service access request is received from the access plane or from the management plane components, (From Service consumers to service providers for incoming request and from service providers to service consumers for outgoing response) by SGS, it is passed on to SDN orchestrator, for provisioning microsegmented services.

The SDN Orchestrator verifies the identity of the access and management plane components involved in the microsegmentation process by passing the request to the Identity proofing module. The identity proofing module validates the request and the result is passed on to the SDN orchestrator.

Upon receipt of authentication and authorization, the SDN Orchestrator provisions VNF instances to microsegmented service, based on the requirement, with adherence to the policy and trust engine. The VNFs of the same microsegmented service are grouped into clusters for ease of management and provisioning. Every cluster is specific to its corresponding microsegmented functionality. First, the orchestrator searches for already provisioned VNF instances within the cluster. This includes both the previously provisioned and the scheduled VNF instances. The VNF instances of same category within a particular cluster are governed by the identity aggregation module of the IdAM proposed above. The access policy, specification of VNF, quality of service parameters including the throughput, is compared with the microsegmented service for aggregation and clustering.

If the orchestrator is unable to use an already-deployed VNF instance, it should start a new VNF instance in a cluster. If the required service specification including policy and trust is satisfied, the algorithm schedules the VNF within the cluster, else the VNF instance cluster should not be taken into consideration and new cluster is formed, as per the policy and trust. Upon completion of all the requests from single service consumer, the microsegmented services are deactivated and VNF instances are deprovisioned by the Identity deactivation module, which is considered as future work. The proposed methodology ensures that the VNF provisioning and assigning microservices are executed without major end-to-end delay.

V. ALGORITHM

The algorithm for Security Orchestration with Microsegmentation is described as follows,

- Input: Service access request (from Service Consumer and Service Provider), Identity validation, QoS parameters, Trust engine output.
- Output: Provisioned Microsegmented services on VNFs.
- Variables:
  - SGS_Request: Service Access Request received from Secure Gateway Service.
  - IdentityProofResult: Result of Identity Proofing.
  - VNF_Cluster: Collection of VNF instances for a specific microsegmented service.
  - Existing_VNF: Pre-existing or scheduled VNF instances within a cluster.
  - New_VNF: New VNF instance to be deployed.
  - Policy_Match: Boolean variable indicating if the policy and trust requirements are met.
- Steps:
  i. Receive Service Request:
     - Receive SGS_Request from the Secure Gateway Service (SGS).
  ii. Identity Proofing:

- Send SGS_Request to the Identity Proofing module.
- Receive IdentityProofResult from the Identity Proofing module.

iii. Authentication and Authorization Check:
- If IdentityProofResult is "Authenticated and Authorized":
  - Proceed to step iv.
- Else:
  - Return "Authentication/Authorization Failed."
  - Terminate.

iv. Microsegmentation and VNF Provisioning:
- Send SGS_Request, Access Policy, QoS Parameters, and Trust Engine Output to the SDN Orchestrator.

v. Cluster Search:
- Search for an existing VNF_Cluster associated with the requested microsegmented service.

vi. Existing VNF Utilization:
- If VNF_Cluster exists:
  - Retrieve Existing_VNF instances from the VNF_Cluster.
  - Send Existing_VNF, Access Policy, QoS Parameters, and Trust Engine Output to the Identity Aggregation module.
  - Receive result from Identity Aggregation module.
  - If result is successful:
    - Utilize the Existing_VNF instances for the microsegmented service.
    - Proceed to step 9.
- Else:
  - Proceed to step 7.

vii. New VNF Deployment:
- Create a New_VNF instance based on the required specifications from SGS_Request.
- Check if the New_VNF instance satisfies the Access Policy, QoS Parameters, and Trust Engine Output.
- If all requirements are satisfied:
  - Set Policy_Match to TRUE.
- Else:
  - Set Policy_Match to FALSE.

viii. Cluster Formation/Scheduling:
- If Policy_Match is TRUE:
  - If VNF_Cluster does not exist:
    - Create a new VNF_Cluster.
  - Add New_VNF to the VNF_Cluster.
  - Schedule New_VNF within the VNF_Cluster.
  - Proceed to step 9.
- Else:
  - Create a new VNF_Cluster with the New_VNF instance, as the policy and trust requirements failed with existing clusters.
  - Proceed to step 9.

ix. Provision Microsegmented Service:
- Provision the microsegmented service using the selected VNF_Cluster.

x. Session Completion and Deprovisioning:
- Upon request completion:
  - Deactivate the microsegmented service.
  - Send VNF_Cluster to the Identity Deactivation module.
  - Deprovision the VNF_Cluster instances.

xi. End:
- Terminate the algorithm.

This algorithm provides a detailed framework for Security Orchestration with Microsegmentation based on ZTA. The proposed methodology ensures that the VNF provisioning and assigning microservices are executed without major end-to-end delay.

VI. IMPLEMENTATION

The Integrated Environment is built using public and private clouds, edge nodes, and IoT gateways. Two Private Cloud environments are being created for Service Providers, the first of which is using OpenStack 16. on two nodes, followed by VMware (17) on two nodes. Amazon Web Services (18) and Google Cloud (19) are being evaluated for public cloud services. StarlingX (20) is used to create an edge computing environment with two nodes. Cloudsim (21) generates a simulated cloud environment. OpenStack Cloud and VMware Cloud are used as Service Consumers. One of the Edge nodes and one of the Cloudsim nodes are designated as service consumers. OpenvSwitch is employed on the Cloud and Edge nodes for implementation of the software defined networking environment, with SDN orchestrator. The VNF instances are deployed as docker containers. NS-3 Network simulator tool (22) is employed for simulation of network. Mininet (23) is employed for OpenvSwitch based SDN network emulator, OpenFGA (24) for authorization, OpenZiti (15) for Zero Trust framework. The workflow with SGS is implemented with Python. The experimentation was performed by comparing the existing approaches with the proposed approach. The result evaluation results are tabulated in table 1.

Table 1 Time for provisioning

S. No.	No. of simultaneous requests	Time taken for new request process by proposed approach in seconds	Time taken for new request process by SGS, SDN Orchestrator, Identity proofing, Identity aggregation, VNF instance provisioning, Clustering for processing in series in seconds
1	1	0.5	0.6
2	5	0.7	0.8
3	10	0.85	1.2
5	25	1.2	2.5
6	50	1.9	3.6
7	75	2.3	4.4
8	100	2.5	5.2

The graphical representation of the above experimented data follows,

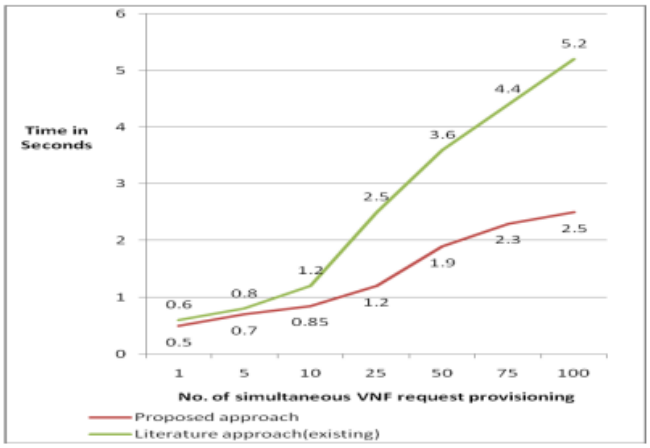

Figure 4 : VNF Provisioning

During the execution of the experimentation, multiple simultaneous microsegmentation requests were passed on to the SGS service. The time taken for provisioning multiple simultaneous the VNF instances, positing in cluster, assigning the VNF instances with microsegmentation services is evaluated. It is observed from the experimentation that the time taken for provisioning the VNF instances increases gradually with a drastic increase in the number of simultaneous microsegmentation services requested. The time taken by the proposed approach is less when compared with the existing algorithms. This ensures that the proposed system, with adherence to ZTA can be deployed on production Clouds and Edge, since the response time is limited irrespective of the number of concurrent microsegmentation requests.

VII. CONCLUSION AND FUTURE WORK

ZTA and a robust security framework, which is overseen by a Security Manager, provide a comprehensive solution for securing the heterogeneous Integrated Environment. The Security Manager, designed and implemented with a foundation in Zero Trust Architecture, demonstrates a robust and adaptable approach to securing a complex, integrated environment encompassing public and private Clouds, Edge nodes, and IoT gateways. By leveraging Identity and Access Management and Security Orchestration with Microsegmentation, the system effectively mitigates risks, overcoming the threats and enhances overall security posture. The IdAM component, encompassing entity discovery, identity proofing, aggregation, and deactivation, establishes a strong identity-centric security perimeter. By rigorously verifying the identity of every entity, whether user, device, or application, the system ensures that only authorized access is granted. The Security Orchestration, utilizing SDN and Virtual Network Function clusters, enables the creation of highly granular microsegments. The integration of Secure Gateway Service as a central communication point streamlines the interaction between service consumers, providers, and the security manager. This centralized control ensures consistent policy enforcement. The proposed framework has been effectively executed on the Integrated environment testbed utilizing open-source technologies. The experimentation concludes that the authentication method is more suitable for practical applications. Zero Trust Architecture constitutes the foundational element of the security dimension in the paper. The paper centers on two primary elements: identity and access management, and security orchestration using microsegmentation. The IdAM module guaranties the authentication and confirmation of all entities within the Integrated Environment. This encompasses both Service Consumers and Service Providers, executed during the operational phase of the environment in accordance with the principles of Zero Trust Architecture, which requires continuous identity verification and validation. Security orchestration utilizing micro-segmentation fundamentally divides the environment into various zones from a security standpoint. This aids in limiting any assault to a more confined area, so minimizing lateral mobility and preventing access for a prospective assailant. VNFs facilitate the execution of microsegmentation by dividing the network into smaller areas with varying security tiers. The Security Manager oversees the VNFs in the Integrated Environment by automating their provisioning, configuration, and lifecycle management throughout deployment. Our suggested approach demonstrates the effectiveness of employing VNFs for this objective. The suggested design coordinated the micro-segmentation operations, therefore decreasing the necessary provisioning time and diminishing or perhaps eradicating human involvement. The approach emphasizes the allocation of nearly ideal configurations in a timely manner.

A solid basis for trust has been created by meticulously tackling obstacles and utilizing strengths; additional research could enhance security inside the interconnected ecosystem. Identity deactivation either suspends, deactivates, archives, or eliminates records based on the outcomes of identity aggregation, in alignment with environmental policy. At now, the deactivation of identity is excluded from the purview of this paper and regarded as a subject for future investigation. Orchestration can flexibly modify microsegmentation protocols in response to real-time threat data or alterations in application conduct. Upon the identification of a danger within a microsegment, security orchestration can facilitate an automated reaction. This may entail segregating the compromised section, preventing harmful traffic, or initiating corrective procedures. This facilitates swift and efficient management of security breaches. IdaM is closely linked with VNF clusters and the security orchestration platform, enabling meticulous regulation of service access inside the integrated environment. VNF clusters to facilitate workload mobility within the Integrated environment. The future will encompass dynamic functionalities of policy and trust engines driven by artificial intelligence, as an extension of existing efforts. Standardizing policies throughout the digital realm would undoubtedly create a significant impact on the Integrated Environment. The Security Manager is capable of interfacing with VNFs to gather security information and scrutinize network traffic for any threats and irregularities. Future endeavors regarding the VNF clusters involve the implementation of security orchestration to oversee the lifecycle of VNF cluster constituents, while adaptively modifying microsegmentation policies in response to the health and performance of the VNF cluster. Future considerations for security orchestration involving microsegmentation encompass logging and error management for each phase, reporting, and the enhancement of search and scheduling algorithms for improved speed.

REFERENCES

[1].Yilong Zheng, Shunfang Hu, Liangxiong Wei, Yanru Chen, Hao Wang, Yanbing Yang, 2020, 'Design and Analysis of a Security-Enhanced Three-Party Authenticated Key Agreement Protocol Based on Chaotic Maps', in IEEE Access, vol. 8, pp. 66150-66162.

[2]. Scott Rose, Oliver Borchert, Stu Mitchell, Sean Connelly, 2020 'Zero Trust Architecture', National Institute of Standards and Technology, NIST Special Publication, pp.800-207

[3]. Ana Hermosilla, Alejandro Molina Zarca, Jorge Bernal Bernabe, Jordi Ortiz & Antonio Skarmeta, 2020, 'Security Orchestration and Enforcement in NFV/SDN-Aware UAV Deployments', in IEEE Access, vol. 8, pp. 131779-131795.

[4]. Chuanyi Liu, Guofeng Wang, Peiyi Han, Hezhong Pan & Binxing Fang, 2017, 'A Cloud Access Security Broker based approach for encrypted data search and sharing,' 2017 International Conference on Computing, Networking and Communications (ICNC), Silicon Valley, CA, USA, pp. 422-426

[5]. Annapurna P Patil, Gaurav Karkal, Jugal Wadhwa, Meer Sawood & Dhanush Reddy, K, 2020, 'Design and Implementation of a Consensus Algorithm to build Zero Trust Model', 2020 IEEE 17th India Council International Conference (INDICON) pp. 1-5

[6]. Xie, L, Hang, F, Guo, W, Lv, Y & Chen, H, 2021. 'A Micro-Segmentation Protection Scheme Based on Zero Trus tArchitecture,' ISCTT 6th International Conference on Information Science, Computer Technology and Transportation, Xishuangbanna, China, pp. 1-4.

[7]. Mengyu Ma, Zhibin Yu, Bozhong Liu, 2023, 'Automatic Generation of Network Micro -Segmentation Policies for Cloud Environments,' 2023 4th International Seminar on Artificial Intelligence, Networking and Information Technology (AINIT), Nanjing, China, pp. 1-5.

[8]. Xiao, W, Zhang, X & Wang, D, 2022, 'Cross-Security Domain Dynamic Orchestration Algorithm of Network Security Functions,' 7th IEEE International Conference on Data Science in Cyberspace (DSC), Guilin, China, pp. 413-419.

[9]. Bringhenti, D, Sisto, R & Valenza, F, 2023, 'Security automation for multi-cluster orchestration in Kubernetes,' IEEE 9th International Conference on Network Softwarization (NetSoft), Madrid, Spain, pp. 480-485.

[10]. Vinoth, R, Lazarus Jegatha Deborah, Pandi Vijayakumar, & Brij B Gupta, 2022, 'An Anonymous Pre-Authentication and Post- Authentication Scheme Assisted by Cloud for Medical IoT Environments', IEEE Transactions on Network Science and Engineering, vol. 9, no. 5, pp. 3633-3642.

[11]. Xiaotong Zhou, Debiao He, Jianting Ning, Min Luo & Xinyi Huang 2023, 'AADEC: Anonymous and Auditable Distributed Access Control for Edge Computing Services', IEEE Transactions on Information Forensics and Security, vol. 18, pp. 290-303

[12]. Miloud Bagaa, Tarik Taleb, Jorge Bernal Bernabe & Antonio Skarmeta, 2022, 'QoS and Resource-Aware Security Orchestration and Life Cycle Management', IEEE Transactions on Mobile Computing, vol. 21, no. 8. doi: 10.1109/TMC.2020.3046968

[13]. Phu Nguyen, Rustem Dautov, Hui Song, Angel Rego, Eider Iturbe, Erkuden Rios, 2023, 'Towards Smarter Security Orchestration and Automatic Response for CPS and IoT,' 2023 IEEE International Conference on Cloud Computing Technology and Science (CloudCom), Naples, Italy, pp. 298-302.

[14]. Aytaj Badirova, Shirin Dabbaghi, Faraz Fatemi Moghaddam, Philipp Wieder, Ramin Yahyapour, 2023, 'A Survey on Identity and Access Management for Cross-Domain Dynamic Users: Issues, Solutions, and Challenges', in IEEE Access, vol. 11, pp. 61660-61679

[15]. https://openziti.io/

[16]. https://www.openstack.org/

[17]. https://www.vmware.com/

[18]. https://aws.amazon.com/

[19]. https://Cloud.google.com/

[20]. https://www.starlingx.io/

[21]. https://github.com/Cloudslab/Cloudsim

[22]. https://www.nsnam.org/

[23]. https://mininet.org/

[24]. https://openfga.dev/

[25]. https://openziti.io/