2021 6th International Conference for Convergence in Technology (I2CT)
Pune, India. Apr 02-04, 2021

Building A Zero Trust Architecture Using Kubernetes

Daniel D'Silva
Electronics and Telecommunication
Sardar Patel Institute of Technology
Mumbai, Maharashtra
daniel.dsilva@spit.ac.in

Dayanand D. Ambawade
Electronics and Telecommunication
Sardar Patel Institute of Technology
Mumbai, Maharashtra
dd_ambawade@spit.ac.in

***Abstract*— In the twenty-first century, trust has become an influential factor in people and organizations. As the world is advancing digitally, mobile and cloud services have become the principal drivers of this era. The conventional frameworks to protect such an environment have dissolved. There existed a period where organization resources were put away inside the secure perimeter and regarded as safe. Moreover, the recent work-from-home culture provides attackers with a rather significant opportunity to breach security controls. Everybody is deemed trustworthy inside the network, allowing an intruder to gain escalated access inside the perimeter. These fortresses currently permit clients to get sustained information from outside the fortification since everybody is 'trusted excessively.' making our current foundation defenseless to attackers. This paper proves 'Zero Trust' as another worldview of online protection. It explores the previous work related to Zero Trust implementation and its research. It discusses Zero Trust as a potential for future network security. It uses containers to implement the architecture, which responds to various types of attacks. It focuses on security at every OSI model layer and the advantages and disadvantages of Zero Trust Architecture.**

Keywords—Zero Trust, Kubernetes, Access Control, Keycloak, Proxy.

I. INTRODUCTION

Networks and technology have taken a significant leap in terms of advancement and automation over the years. However, the job of ensuring their safe communication without getting attacked has been a significant concern. In the early days of the internet, organizations used to store their data within the enterprise, thus creating a perimeter. This network topology was highly fortified. Thus, it was not easy for an attacker to breach the organization's infrastructure and assets. Technology has allowed us to access data from outside the enterprise space's four corners in recent years.

In 2004, before Zero Trust had uniqueness, the idea of deperimiterization was initiated by Jon Measham and promoted by the Jericho Forum from OpenGroup. The information security officers at this organization developed the Jericho Forum Commandments, which established the areas and policies observed while looking at a de-permiterized future. Paul Simmonds from International Consults and Investigation (ICI) had determined that there is a need for a new security model. He decided that de-permitierization is a concept that solves enterprise needs without a tightened perimeter, which would bring rise to the potential for new opportunities. The author's thinking brought in the era of a renewed model called 'Zero Trust.' first introduced by Forrester Research analyst John Kindervag becoming the beginning of a Zero Trust. [1]

Zero Trust is a cybersecurity paradigm focused on resource protection and the premise that trust must not grant implicitly [2] and frequently evaluated. The current infrastructure relies on a solution that helps restrict privileged access, and its core is Privileged Access Management (PAM). It merely works by limiting access to a particular service by account authentication and authorization. Should a user need access to a domain, he is not permitted; the user must request authorization and finally get the approval. Legacy Privileged Access Management's design was to work for systems and resources inside an enterprise or organization network. The framework executives would have a solitary independent 'root' account that they would use from a secret vault [3] to get to the organization's assets. Those mentioned were safe when no data or resources, or credentials needed access from outside the network. However, the upsurge in today's generation and the introduction of cloud computing have made PAM challenges. Fifty-three percent of organizations host at least half of their cloud infrastructure. [4] More and more data and resources get used from outside the network than inside it.

An attacker can easily compromise the entire system by gaining access to the network, masquerading themselves as legitimate users to access control policy. Perhaps through social engineering, a trusted user's negligence, ignorance, or innocence. After that, it is as simple to reset passwords and gain nearly unlimited access. [5] Despite spending an estimated $137 billion on various security technologies in 2019, two out of three enterprises experience data breaches at an average of five breaches per organization. [3] In 2015, a fifteen-year-old British kid effectively hacked his way into the records of CIA boss John Brennan, FBI chief Mark Giuliano, and US Homeland Security Secretary Jeh Johnson. He could take government insight reports, reset staff iPads, and show provoking messages on Johnson's home TV. [5] Hackers from a foreign land attempted 40,300 cyber attacks on India's Information Technology infrastructure and banking sector [6] in about five days in June 2020. According to CISCO Annual Cyber Security Report 2019, Eurofins Scientific, a forensic firm used by police forces across the country, suffered a massive targeted ransomware attack. The firm deals with more than seventy-thousand criminal cases every year. Due to the cyberattack scale, several court cases were made to be adjourned. [7] According to CISCO Annual Cyber Security Report 2018, fifty-three percent of all cyberattacks led to more than $500,000, including, but not limited to, lost revenue, customers, opportunities, and out-of-pocket expenses. [4]

2021 6th International Conference for Convergence in Technology (I2CT) | 978-1-7281-8876-8/21/$31.00 ©2021 IEEE | DOI: 10.1109/I2CT51068.2021.9418203

Many of the organizations are unaware of how the breach has occurred [8].

Zero Trust aims to protect people, property, and infrastructure (PPI) from attackers that can potentially threaten enterprise or organization data. As civilization evolves to connect through technology's inevitable ubiquity increasingly, securing systems, networks, and data on which we rely has become pre-eminent. [5] This paper gives a detailed summary of Zero Trust, its evolution, where it stands today, and how it reshapes the future trust landscape.

II. Related Work

Advancement of research in networking has brought about the current infrastructure all of us live in and network with each other; however, a fundamental property under jeopardy is trust. Now, let' define trust. As per the Oxford English Dictionary, it means to have confidence in somebody and believe that somebody is good, sincere, honest. [9] With the existing infrastructure, it has been nearly impossible to differentiate between trusted and untrusted interfaces. A lucrative opportunity for hackers is that trust does not apply to packets essentially means that IP and MAC address perhaps are candidly exposed through a packet sniffer. Moreover, packets cannot trust, and likewise, network engineers cannot trust them.

Focusing mainly on trust and inserting into the minds of network engineers, the authors [12] proposed a model that would essentially revolutionize the past decade and make various companies, most notably Google, rethink their network infrastructure and opt for a relatively advanced one. The model proposed is called 'Zero Trust.' A model that trusts nobody from the inside or trusted network, and the external or untrusted network. By default, it assumes that the attacker is present on the network and deems all network traffic untrusted. The first idea of Zero Trust was an information-driven organization plan that utilized micro-segmentation [13] to enforce more granular guidelines and limit the attack possibilities. Since its beginning, the idea of Zero Trust and its advantages have developed essentially. These days, Zero Trust is being utilized by associations to drive key security activities and empower business chiefs and IT pioneers to execute soberminded anticipation, discovery, and reaction measures. The initial mantra of Zero Trust proposed by [12] was 'never trust, always verify.' The researchers in [3] have renewed this mantra to 'never trust, always verify, enforce least privilege.'. Unfortunately, this is still a common practice. A study revealed that sixty-three percent of responders mentioned that their companies usually take more than one day to shut off privileged access for employees who leave the company.

One of the critical steps of Zero Trust is to have multiple steps to authenticate a user. While executing multi factor authentication (MFA), one must authorize National Institute [3] for Standards and Technology (NIST) Authentication Assurance Level 2 (AAL2), characterized in NIST Special Publication (SP) 800-63 for all administrators. NIST AAL2 requires "possession and control of two distinct authentication factors": something one must know and have. A good example is a password combined with a push notification to a user's smartphone or a one-time password (OTP) generated by your smartphone. [3] For critical assets, NIST AAL3 is recommended, where possible. NIST AAL3 requires proof of possession of a hardware-based cryptographic token, such as a smart card or FIDO key. Those authenticators can then be used in combination with a password or personal identification number (PIN). Access to admins must be given through an administrative jump box [3].

The author mentioned some pivotal points that define the 2020 Zero Trust model, such as; using a network segmentation gateway (SG), designed to be the nucleus of the network compared to the existing unified threat management system (UTM). Its job is to combine multiple standalone security products of the existing infrastructure and act as one central module. The SG must handle a 10 Gigabit connection while providing Quality of Service (QoS) to maintain performance. Having an SG would mean it must define the global policy and enforcement rules. Zero Trust would require the network divided into switching zones. Another new interface suggested that is mandatory is Microcore and Perimeter (MCAP). The job of an MCAP is to manage the zone and the resources within the area. As discussed earlier, every zero trust model must have its data logged. That is the job of an all-new network called the Data Acquisition Network (DAN). DAN's function is to have a log of the network and analyze it in real theoretical time.

Three properties that define every Zero Trust network are (i) All resources need to be accessed securely, regardless of their physical or logical location, (ii) Have stringent access control policies, and finally (iii) Capture and Log all network traffic. [14]

Fig. 1. NIST Concept of Zero Trust

Figure 1. describes a concept of Zero Trust by NIST. The resource, that is, System, Data, or Application, and the client, a middle man, acts as a proxy between the two. The proxy or broker path is assumed to be untrusted, albeit the path between the proxy and the resource must be an 'absolute trust zone'. Wherein all the broker's inputs are trusted; however, it still needs to be checked. The broker's role is to subjugate two primary functions, i.e., Policy Decision Point (PDP) and Policy Enforcement Point (PEP). [15] The part of PDP is to ensure that the traffic flowing under it is trusted. In hindsight, zero trust provides protocols and conceptualizes moving the PDP and PEP closer to the resource. It would specifically authenticate and authorize all subjects, asses, and workflows that make up the enterprise. [15] The authors define a hypothesis concerning zero trust:-

1. Every data source, as well as computational services, are recognized as resources.
2. Communication, in any form, must be secured, notwithstanding network location.
3. Every resource within the enterprise must be granted solely on a session basis and regulated by a policy.
4. The enterprise must monitor and measure the integrity as well as the security posture of all assets.

5. Every resource must be dynamic and scrupulously authorized and authenticated before access to it is allowed.

As always, when it comes to network planning and deployment, there are assumptions to be made. For Zero Trust Access, these are the following assumptions:

1. The local area network inside an enterprise should not be considered as an implicit trust zone.
2. With the recent trend of bring-your-own device (BYOD) implemented in enterprises, it is assumed that devices being connected to the network are not an entity of the enterprise since any device can be compromised.
3. Resources are never trusted, i.e., from a security standpoint, every asset or resource must be continuously evaluated and must only be subject to use as long as it is needed.
4. Cloud services have become an essential part of every enterprise network, making it evident that all the enterprise resources are not inside the enterprise-owned infrastructure.
5. All connection requests outside the enterprise, such as Remote Desktop, must be authorized and authenticated. All data must be communicated with respect, confidentiality, integrity, and source authentication.
6. Based on the assumptions mentioned above, the crucial one is that all assets and data communications between enterprise and non-enterprise infrastructure must continuously be under security strategy and stance.

III. Zero Trust: The Approach to Redefining Cybersecurity

Zero Trust is a cybersecurity paradigm that trusts nobody, no device, and no application yet supports all of them by periodically verifying their authenticity and authority. A Zero Trust Architecture (ZTA) can either be implemented over an existing infrastructure or wholly redesigned from the ground up. The scope of such an architecture is to provide uttermost security keeping in mind the safeguarding of all assets under Protected Personal Information (PPI).

Three components of Zero Trust architecture are user and application authentication, device authentication, and most importantly, trust. Unlike the existing infrastructure, wherein a user is authenticated just once. A ZTA keeps checking the user's authenticity, monitors the user's devices, and checks for any location change initiated by the user device. Moreover, it also regularly checks for any discrepancies in the application that the user would be using. Should there be any form of alteration, the architecture must terminate the connection with immediate effect. In the case of any data manipulation, data is restored through the backup while keeping logs of every minuscule activity.

IV. Proposed Model

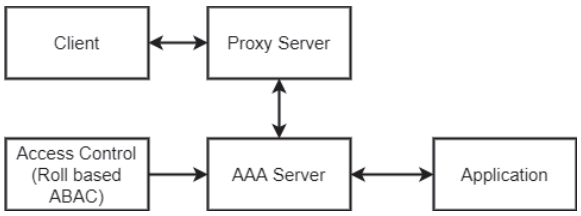

Fig. 2. Block Diagram of Proposed Architecture

In this section, we propose a Zero Trust model for a cloud computing environment with practical experimentation. In this era, where everyone is accessing information outside of the organization, cloud-based services have increasingly become a security pinnacle. The existing or rather traditional organization, based around a perimeter, fails to provide user and application security. Zero-Trust is fit for cloud-based services and network security within the organization since it trusts nobody and no service. A Zero Trust strategy enforces strict and specific access control to advance cloud security while maintaining records or logs of every activity within the network.

Figure 2. shows the system architectural overview of the work conducted. It can be seen that the client connects to the proxy server. The proxy server configuration to be a reverse proxy; hence the user does not know the real IP address of the Authentication and Authorization Server. The proxy server then redirects the client to the Authentication and Authorization Server. Access Control decides if the user is allowed or denied access to the application. Once the user is successfully authenticated, he has the authorization to access the application. The authentication and authorization server continually checks for certificates to ensure that no compromised user enters the system.

A. Client

The client is any client having a web browser.

B. Proxy Server

The proxy server is responsible for passing the request from the client to the Kubernetes cluster. It must be a physical machine having the ability to reverse proxy. The proxy server chosen for performing this work Is Squid Proxy version 4.13, an open-source proxy server with caching.

C. AAA Server

The architecture's core block is the authentication, authorization, and application server. It acts as the only mediator between the proxy server and the application. At the heart of this server lies Kubernetes, an open-source platform for managing containerized workloads and services. [16] Every application in this architecture is inside a container. This server doubles up to manage containers and is given a second name, the Kubernetes Master Server. A container is a standard unit of software that packages up code and all its dependencies, so it runs quickly and reliably from one computing environment to another. [17] Applications that are in containers are the frontend React JS, the back-end application SQL database. The work implements Keycloak, a containerized authentication, and authorization tool for devices and clients.

D. Access Control

The Access Control block is built within the application and authorization server but deserves its place exclusively. This work implements Role-Based Access Based Access Control, a hybrid of the traditional and overused RBAC, and the new advancement of ABAC access control. It means that a client is given the authority he has within an organization but is given a specific attribute within that authority.

E. Application

The application is clustered within the Kubernetes cluster. The applications sit on another virtual machine together. However, they are connected to the Kubernetes via permanently assigned bearer tokens, making them a slave of the Kubernetes master node.

V. Inside the AAA Server

The following are the tools and services used to implement this architecture:

1) Lightweight Ubuntu v20.4.1
2) OpenID Connect v1.0
3) Kubernetes v1.19.3
4) Docker v19.03
5) Keycloak v11.0.3
6) React js v17.0.1
7) Nginx v1.18.0
8) Squid proxy v4.13
9) Mozilla Firefox v82.0.3

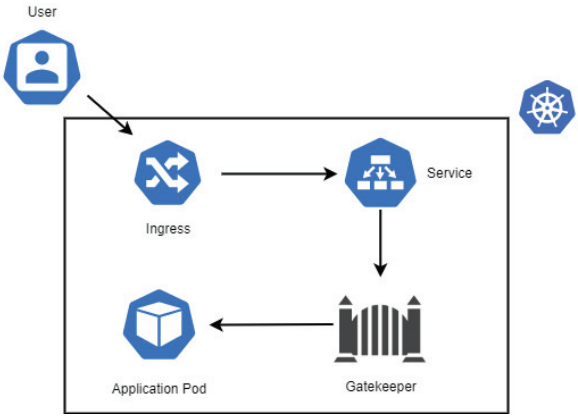

Fig. 3. Authentication and Authorization Process

The entire work is done on Ubuntu, a Linux distribution. We use control groups to constrain resources allocated to processes. Systemd, is the default init system for a Linux distribution. However, the init process generates a root control group called cgroup and acts as a cgroup manager. Systemd has tight integration with cgroups and allocates a cgroup per systemd unit [16].

The application server and authentication server use Docker CE, since it is most compatible with Kubernetes.

When the request is passed from the proxy server to the Authentication Server, it is welcomed by the Kubernetes Ingress. The Kubernetes Ingress, controlled by the NGINX Ingress Controller, is configured to act as a load balancer for multiple requests coming into the Kubernetes cluster.

Kubernetes Ingress then forwards the request to the exposed authentication service. In this case, Red Hat Keycloak is given the request. Keycloak is configured to use Gatekeeper, an adapter that integrates with the Keycloak authentication service. Gatekeeper is a sidecar container deployed on the Kubernetes pod. A pod is simply an instance of a process actively running. Our application, the webpage, first points to Gatekeeper rather than to itself. This creates a natural proxy for incoming requests. Gatekeeper is responsible for communicating with Keycloak about user credentials. Should a user be already logged in and Keycloak gets a request to login again, it logs out from the previous session and asks to re-validate the user.

We chose to implement XACML (eXtensible Access Control Markup Language), developed by OASIS (Organizaton for the Advancement of Structured Information Standards), developed for user authentication. [18] XACML is an attribute-based access control (ABAC) system. An attribute is given to the user that decides whether a user has access to a given resource. RBAC is implemented, but as a specialization of ABAC.

1) Policy Administration Point (PAP): PAP provides a UI based on the Keycloak Administration Console to manage the resources, scopes, permissions, and policies.

2) Policy Decision Point (PDP): The PDP provides a distributable policy decision point to where authorization requests are sent. Policies are evaluated according to the requested permissions.

3) Policy Enforcement Point (PEP): PEP provides implementations for different environments to enforce authorization decisions at the resource server-side [19].

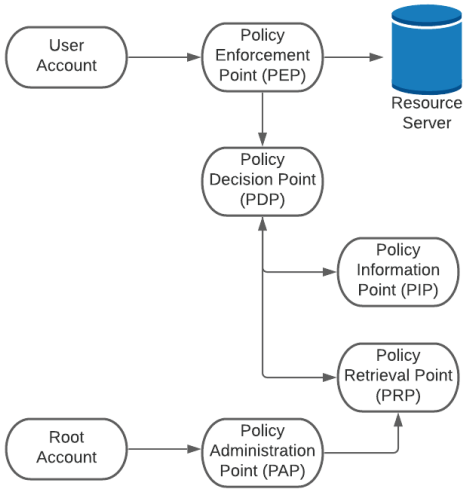

Fig. 4. XACML Working

For any Kubernetes communication, the work uses OpenID Connect, which incorporates OAuth 2.0. Every service account, such as the Kubernetes Dashboard, is verified with OAuth. OAuth protocol works on the fundamentals of requesting an access token from the Kubernetes Authorization server. The auth server then responds with an access token and id token, combined to make JWT (JSON Web Tokens). [20] The service account then uses the JWT to access the API server. These tokens expire as per a set amount of time.

This architecture also uses Single Sign On (SSO) to authenticate users with various applications only by a

specific credential. Given the example of a Google Account, the flow looks like this:

1) User browses to the website or application hosted.

2) The user is redirected to Keycloak, who then sends back a token containing some information to the SSO, simply known as the Identity Provider.

3) The Identity Provider checks to see if the user has previously been authenticated. If a user is logged in, the following step is ignored.

4) If a user is not logged in, the user is prompted to provide the identity provider's username and password.

5) Once the Identity Provider validates the credentials, it sends back a token to Keycloak confirming that authentication is a success.

6) The token is returned to the service provider through the user's browser.

7) If a token received is legitimate, the user is granted access to the website's resources.

It is essential to keep every minute activity recorded. It is recommended to use every component in the architecture to log data. Kubernetes and Keycloak both maintain logs periodically of every small activity, essential for Zero Trust.

Figure 5 demonstrates the flow chart of the implemented work. The algorithmic flow is mentioned in the steps below:

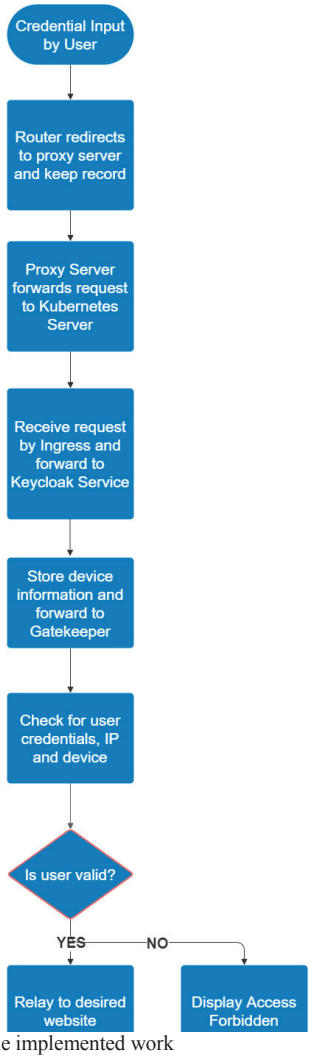

Fig. 5. Flow chart of the implemented work

1) The client requests a particular web page by typing the Domain Name System (DNS) Name.

2) The proxy server keeps track of the client page who has requested the page and forwards the Kubernetes cluster request.

3) Within the Kubernetes cluster, the Ingress accepts the request and forwards it to the Keycloak and Gatekeeper service.

4) Keycloak then validates the user through basic authentication such as ID and Password and verifies the machine's authenticity through X.509 certificates.

5) Once the user is verified, it informs Kubernetes to redirect the request to the application.

6) During this time, the Kubernetes cluster keeps track of the certificates and continually checks their authenticity.

VI. Response to Attacks

There are various types of cyber attacks. These attacks are mainly classified as follows:

A. Infection based attacks

These attacks are executed through malicious code written and installed in a user's computer through Phishing, Malvertising, and Drive-by downloads. The repercussions of such attacks are identity theft, data being stolen, and privileged access.

Before an attacker can infiltrate the network, he/she has to pass the security perimeter. The main focus of Zero Trust is to create a secure perimeter. Should an attacker enter employee credentials that the attacker obtained by social engineering, the attacker is limited by the simple two-factor authentication, used while deploying this infrastructure.

Suppose the attacker can, for some reason, get the OTP or PIN of a legitimate employee. In that case, RBAC-ABAC's combination proves to be difficult for the attacker to penetrate due to unprivileged access given to the employee. If an attacker gets credentials, let us say its CEO account and try to copy data from a database or manipulate it. The role of that account is to view and not edit. The automatic backed-up data must be live.

B. Explosion attacks

Explosion attacks are exploits made in the system due to previous ignorance. Examples of such attacks are buffer overflow attacks wherein an attacker exploits an application's memory, which results in changes in the main execution path, leading to damage of files or critical information.

Assuming that an attacker passes the secure perimeter, the second line of defense against such attacks is the application of health monitoring within Kubernetes. Should any package not be updated, that causes a buffer overflow. The pod gets immediately destroyed if any discrepancies occur, thus keeping at bay explosion attacks.

C. Probe Attack

A probe attack, commonly known as a sniffing attack, is the continual monitoring of traffic on a or multiple ports to look out for an opportunity to attack the network. Probe attacks, for example, are done on layer 7 of the OSI model.

To defend against probing attacks, it is ensured to close all ports in Kubernetes using Network Policies. If a port needs to be open, it remains open for as long as the session is active.

D. Cheating Attack

An attack is called a cheating attack when an attacker impersonates a genuine user. Such attacks are more commonly known as cheating attacks. The methodology to counter such attacks is the same as a probing attack.

Suppose an attacker spoofs MAC address, a typical implementation done for cheating. In that case, the architecture tracks users based on IP address location as a security measure. The result of which is a comparison as to how did a legitimate user change location so quickly. Hence, this will also be logged and raise the alarm to the root. Thus ensuring that even if an attacker tries to probe, he is suppressed.

E. Traverse Attack

Traverse attacks are, if not the most common types of attacks on any network. A traverse attack is a brute force attack wherein an attacker will submit a skew of passwords having faith that either one of the passwords is correct. Attackers fundamentally also change the header agents during this instance.

Changing header agents or an IP address triggers an alarm in the authentication service. It keeps track of every logged-in user activity in the network. Currently set to the limit of two, if an IP address or header agent is changed more than twice, even a legitimate user is temporarily suspended.

F. Concurrency Attack

A concurrency attack is a type of attack wherein a user transmits concurrent rapid packets of data to temporarily compromise all the users trying to access a particular service. These attacks can first be detected at the Hardware layer (Network Layer) or Transport Layer.

The proxy server is well equipped to handle such concurrent attacks by dropping rapid packages with minuscule flood drop thresholds.

VII. Layered Security

This section discusses the work securing the asset at various layers of the Open System Interconnect (OSI) model.

A. Application Layer

The seventh layer of the OSI model refers to applications that support end-user functions. Here, the most common form of authentication is a username and password. [21] This work emphasizes multifactor authentication (MFA). A user needs to be authenticated with a username and password and a simple One Time Password (OTP). Another implementation of MFA, which was discussed and not proposed, is PIN. A secure PIN alongside a username and password proves vital. However, since it is difficult to get an OTP, it was chosen. A combination of username and password and either one of the two authenticates the user and grants the user access.

Password length and complexity are essential components of an accounting policy. [21] Nowadays, there is an urge for stronger passwords by organizations to employees. Organizations recommend incorporating a combination of Uppercase, lowercase letters, numbers, and special characters.

Password length and complexity are essential components of an accounting policy. [21] Nowadays, there is an urge for stronger passwords by organizations to employees. Organizations recommend incorporating a combination of Uppercase, lowercase letters, numbers, and special characters.

A user can choose to keep their password unchanged indefinitely. However, a user gets a prompt on whether he or she would like to update their password. Finding a balance is critical between user productivity and an appropriate level of security. [21]

It is ensured that every package is up-to-date as outdated packages are one of the main reasons for a cyber breach during this work.

A layer seven load balancer distributes requests based upon data found in this layer protocols, such as HyperText Transfer Protocol (HTTP) and HTTP Secure (HTTPS).

B. Presentation Layer

The Presentation Layer ensures that data is in a usable format and encrypts the data. Keycloak is responsible for encrypting data using XACML, OpenID Connect tokens, OAuth 2.0, and JSON Object Signing and Encryption or JOSE specifications. The reason for selection is because a sophisticated encryption algorithm makes it difficult to gain access.

At the Presentation layer is where we also check for authorization or access control. With Keycloak, a combination of Role-Based Access Control (RBAC) and Attribute-Based Access Control (ABAC) is implemented. It ensures maximum access control with minimal chances of gaining privileged access.

At this layer, the work uses Transport Layer Security (TLS) certificates signed by a Certificate Authority (CA) controlled by us. These certificates and CA are used to establish trust. For the most part, this work uses tokenization. However, we use CA and Certificate Signing Requests (CSR) for service accounts such as the Kubernetes Dashboard and the ‘root’ account. For user accounts, XACML authentication uses a set of policies also defined by us.

C. Session Layer

The session layer is responsible for maintaining connections, controlling ports, and sessions. Kubernetes and Keycloak are configured to ensure that the connection is safe and not tampered with. Should Kubernetes find out that there is a discrepancy or unusual connection between the client and the application pods, it immediately terminates the session. Keycloak, on the other hand, continually checks for X.509 certificates, and should there be any modifications, it also immediately terminates the connection.

D. Transport Layer

The transport layer, also known as the OSI model’s heart, is responsible for bridging the gap between the previously mentioned software layers and the forthcoming hardware layers. Securing this layer is a must. One of the essential components of determining security on this layer is the proxy server. The proxy acts as a natural firewall or De-militarized

done (DMZ) between an untrusted client and the secure network. It is ensured to disable any open ports made known to us by the VAPT test at this layer. Keeping the transport layer security is essential since it is an open gateway to trojan and other viruses. Layer 4 also provides the ability to control traffic, by not just IP and Mac Address of the lower layers, but also by specific applications incorporating the OSI model's upper layers. [21]

A secondary proxy server, commonly known as the Load Balancer (LB), distributes application traffic across several Kubernetes cluster services. Kubernetes Ingress is configured to be an LB service. [21].

E. Network Layer

In this layer, the router is responsible for forwarding the untrusted user request to the Zero trust Architecture. However, inside the Kubernetes cluster, Calico is responsible for defining network policies and acts as a frontline before the data is sent to the upper layers.

F. Data Link Layer

The type of connection made to the architecture, be it wired or wireless, makes no difference since all the data conjugates at Layer 4.

G. Physical Layer

Should this model be deployed, the most prominent way of securing this layer; is by using redundant power supplies, redundant NIC cards, and redundant Ethernet cables to ensure immediate availability at a time of failure.

VIII. Advantages and Disadvantages of Zero Trust

A. Advantages

[22]

1) *Strong policies for user authentication and access:* A Zero Trust Architecture ensures strong management of users inside its network, thus making their accounts secure. Using two-factor authentication or MFA is an optimal way to keep accounts safe. Using a combination of access control policies can ensure minimal compromise to grant access to a specific task.

2) *Data Segmentation*: In a ZTA, a big chunk of data is segmented into types, sensitivity, and use case, which provides additional security. It, in turn, limits users to access the data given for the tasks assigned to them.

3) *Lesser Chance of Vulnerability:* Based on the above two features, there is a much lesser chance of having vulnerabilities leading to attacks.

4) *Tight data protection:* Zero trust keeps data protected during the exchange of information, as well as storage. That includes automated backups and tightly encrypted message transmission.

5) *Excellent security orchestration:* Much like container orchestration, data orchestration is securing all elements; while making them work together efficiently and effectively. ZTA must leave no open vents so that it is nearly impossible for adversaries to penetrate.

B. Disadvantages

1) *Tedious effort and time consuming:* Suppose an organization is upgrading to even a partial ZTA. Making more robust policies and reorganizing them can be challenging since the network must be active and functioning during the transition. Rebuilding the network from the ground up seems like a much easier solution.

2) *Versatile management of dynamic users*: Users need to be monitored at every activity they perform. One user cannot gain access to another attribute. Moreover, users who are not employees must not have special access to the network. Hence, policies need to be redesigned and be attribute and role-specific.

3) *More devices to cater:* In the era of digital devices, being at everyone's table, desk, and pockets, managing devices has become challenging. Users do not have one device but plenty of devices. Each device has its hardware and software properties, its exclusive communication protocols, all of which need to be monitored.

4) *Complex application management:* Applications nowadays are not just a web server but multiple servers and software, each serving its purpose. Some interact with thirdparty applications as well. Keeping such applications in mind, a Zero Trust Architecture must be planned, monitored, and exclusively designed for such needs.

5) *Meticulous Data Security:* With user data stored at multiple locations, each location needs to be well guarded. Every piece of information stored must also be secured with the highest security standards and framework.

IX. Discussions

Zero Trust has a whole, has no governing authority. Hence we believe that plenty of changes can be made in the upcoming years. This work has focused on integrating Zero Trust with the existing security infrastructure.

Concerning the existing infrastructure, Zero Trust proves a cutting-edge security paradigm with hardline policies to ensure that no asset remains compromised. The existing infrastructure still uses the traditional Role-Based Access Control to assign roles. However, RBAC is acceptable; it proves to be inadequate since it is easy to earn privileged access. Zero Trust relies on a combination of access controls, thus ensuring difficulty for attackers to penetrate the perimeter.

Containerization is the future of the cyber spectrum. Everything from applications to authentication is in the development of containers. Keeping this in mind, it is needed to develop infrastructure as per such norms as containers can isolate themselves and are managed by a master orchestrator.

However, everything cannot be software. Physical firewalls and proxy servers play an essential role for years to come. Shifting to Zero Trust is a bold transition and one that takes time. Organizations have progressed to make soft versions of a ZTA implementation on their networks. We see Zero Trust as the future of the internet and cloud computing.

The implementation of the work done is available in [23].

X. Conclusion

This paper discussed micro-segmentation through containerizing applications and implemented it in a Zero Trust Architecture (ZTA) using Kubernetes. We chose to follow the fundamental guidelines given to us by various leading organizations and researchers. We developed our

architecture to enhance the future cybersecurity paradigm. It is discovered that ZTA provides a more robust architecture to redefine cybersecurity. Moreover, with Single Sign-On's rise becoming more and more popular, it was critical to implement it in our architecture.

Other alternatives to Keycloak, such as the Gluu server for authentication and authorization, can also be implemented, keeping in mind the use case. We believe that this architecture needs more research, most notably, the use of HashiCorp Vault for more secure access control, tokens, passwords, and certificates. It is important to note that this model is opensource and needs regular updating, as discussed earlier.

REFERENCES

[1] Andrew Goodman, What Is Zero Trust?. Accessed on: June 25, 2019. Available: https://dzone.com/articles/what-is-zero-trust

[2] Scott Rose, Oliver Borchert, Stu Mitchell, and Sean Connelly, 'Zero Trust Architecture,' NIST, DOI: https://doi.org/10.6028/NIST.SP.800-207 .

[3] Lawrence Miller and Torsten George, Zero Trust Priviledge for dummies, Special Edition, 2019, Centrify, Accessed 29th October, 2020.

[4] Cisco 2018, Annual Cybersecurity Report, Accessed 16th September, 2020.

[5] Pandey, Praful and Mishra, Srishti and Rai, Pooja and Anand, Abhineet," Social Engineering and Exploit Development", International Journal of Scientific Research in Computer Science Applications and Management Studies IJSRCSAMS Volume 8, Issue 5 (September 2019).

[6] https://www.dnaindia.com/india/report-40000-cyber-attacks-attemptedby-chinese-hackers-on-indian-banking-it-sector-in-five-days-2829381, Accessed on: 1st October, 2020.

[7] Cisco 2019, Threats of the Year, Accessed on: 19th August, 2020.

[8]] CGI 2013,' Developing a Framework to Improve Critical Infrastructure' Cybersecurity, NIST.

[9] Oxford Advanced Learner's Dictionary. Accessed on: 15th July, 2020. https://www.oxfordlearnersdictionaries.com/definition/english/trust,

[10] John Kindervag,' Clarifying What Zero Trust Is – and Is Not.' Accessed on: 29th August, 2020. Available: https://blog.paloaltonetworks.com/2018/08/clarifying-zero-trust-not/

[11] "74% Of Data Breaches Involve Privileged Credential Abuse." Accessed, 24th August, 2020. Available: https://www.itsecurityguru.org/2019/02/26/74-of-data-breaches-involveprivileged-credential-abuse/

[12] No More Chewy Centers: Introducing the Zero Trust Model of Information Security," Forrester Research, Tech. Rep., 2010.

[13] S. Mehraj and M. T. Banday," Establishing a Zero Trust Strategy in Cloud Computing Environment," 2020 International Conference on Computer Communication and Informatics (ICCCI), Coimbatore, India, 2020, pp. 1-6, doi: 10.1109/ICCCI48352.2020.9104214.

[14] J. Kindervag, "Building Security into Your Networks DNA: The Zero Trust Network Architecture," Forrester Research, Tech. Rep., 2010.

[15] Zaghdoudi, Bilel and Kaffel-Ben Ayed, Hella and Harizi, Wafa, "Generic Access Control System for Ad Hoc MCC and Fog Computing," Springer International Publishing, 2016, pp. 400-415.

[16] Kubernetes Documentation, Accessed on: 1st August, 2020. Available: https://www.kubernetes.io/docs ,

[17] Get Started with Docker Accessed on: 1st August, 2020. Available: https://www.docker.com/resources/ ,

[18] Altice Labs White Paper,' Identity and Access Management', December 2014. Accessed, 25th October, 2020.

[19] Keycloak - Documentation, Accessed on: 28th July, 2020. Available: https://www.keycloak.org/docs/

[20] Prabath Siriwardena. 2014.' Advanced API Security: Securing APIs with OAuth 2.0, OpenID Connect, JWS, and JWE.' Apress.

[21] Kari A. Pace,' A Layered Security Model: OSI and Information Security', Global Information Assurance Certification Paper, Accessed 29th November, 2020.

[22] K.K. Tucker, Pros and Cons of the Zero Trust Model. Accessed on: 12th, November, 2020.

Available: https://www.infusedinnovations.com/blog/secure-intelligentworkplace/pros-and-cons-of-the-zero-trust-model

[23] https://github.com/dannyboydsilva/Zero-Trust-Network-With-Kubernetes