International Journal of Scientific Research and Engineering Development-– Volume 7 Issue 2, 2024

Available at www.ijsred.com

RESEARCH ARTICLE OPEN ACCESS

Endpoint Security and Surveillance using Zero Trust

Prasanna N[1], Darshan M[2] ,Dhilip N[3] ,Durga Prashanth N[4],Govardhan AV[5]

1(Asst Prof. ,Dept of CSE, KSSEM,Bengaluru,
Email: prasanna@kssem.edu.in)
2 ,3,4,5(Dept of CSE, KSSEM,Bengaluru,)

----------**********----------

Abstract:

Three distinct technical security concepts such as privileged access, strong authentication, and end-to-end encryption with threat detection, must be implemented in order to implement a zero trust strategy. Implementing a zero trust security strategy to the modern architecture by enabling Endpoint protection, enabling Identity to Endpoint Network, Health Monitoring, and Validating Application Security. Threat detection and infrastructure assessment Users receive instant notification.

Keywords — Zero Trust, Endpoint

----------**********----------

I. INTRODUCTION

In order to overcome obstacles, the business model of the organization must change to the "Zero Trust Strategy," a new security approach that does not rely on specific boundary perimeter tools and services. Instead, the Trust model here addresses the special application of authorization and authentication techniques as well as network protection to all devices, procedures, and trust boundaries.

II. PRINCIPLES OF ZERO TRUST STRATEGY

a. Uninterrupted monitoring and Detail verification

The login and endpoint performance update task need to be actively performed to administrator on schedule with detailed logs ensuring authentication and authorization implementation across the boundary.

b. Least Privilege

As the users are always restricted towards the sensitive boundary to maintain results in minimal attack.

c. Importance to Multi parts

The zero trust ensures to categorize endpoints with multiple borders classification using segmentation across network to maintain separate boundary level across the network

d. Preventing go before lateral movement

As an advantage of segmentation zero trust strategy prevents all unintended user moving forward towards routing from the access point to destination endpoint. Since the trusted boundary is maintained across entire network, clearing of the security advisor in each segment is an extreme work to the attacker. And the added feature like dynamic segmentation in network always restricts the attacker from accessing access network endpoints.

e. Multi Factor authentication

The largely building crime evidence stack across the various environment is never decreased. One authentication technique like password is no longer acceptable to trust users and endpoint. So, the demand needs to be upgraded by reinvestigating or authenticating each endpoint with additional chain of user evidence. The Zero trust here mainly coined to implement Multi Factor authentication in network practice to be completed.

f. No Service Integration

The tracking of process across all the endpoint never compromised in zero trust architecture, the visibility of service origin and organization of these service is critical factor and need to be clearly distributed. With this intention the zero-trust strategy usually accepts

III. BUSINESS OBJECTIVE

The main objective of the project is to implement Zero trust security strategy and increase endpoint access defence and achieve network protection by securing traffic, data, and infrastructure. Upon the Zero trust Strategy the building cyber security eco system as follows:

a. Creating a Centralized dashboard for threat to analyze the threats and vulnerability

The centralized dashboard operates based on metrics collected across each endpoint and these collected metrics provides an oversight over endpoints to enable larger insight over endpoint security and these solutions further provide advanced monitoring feature like access management, identity management and endpoint health management as a one-point unified solution to administrator to concentrate over long-term security goals.

b. Reducing attack service to applications using continuous monitoring agent

The agent would be installed on all endpoints and these endpoints agents are responsible for tracking health, performance and also targets towards application security over an endpoint with major task in sharing the endpoint analysis to centralized dashboard on schedule basis.

c. Triggering endpoint identification status and smart notification to admin user

An alarm is notified to the administrator based on the change of device state and activity notification through either by Email, SMS or through third party ticketing application based on the requirement of the user.

d. Enabling multifactor authentication to endpoints

The role of multi factor here doesn't limit the security towards endpoint but also the multi factor authentication need to address the API and cloud service access issues like network security failures and user falsification, so one stricter authentication like One time password, g-auth authentication or mobile user approval is followed across both endpoint and centralized dashboard.

e. Auto Backup solution to increase Application Security

As an overview the trusted backup stored in cloud drive is upgraded with strong encryption with safe storage of secured key. Each file can be analyzed whether to include or not to include in backup process based on the requirement easily.

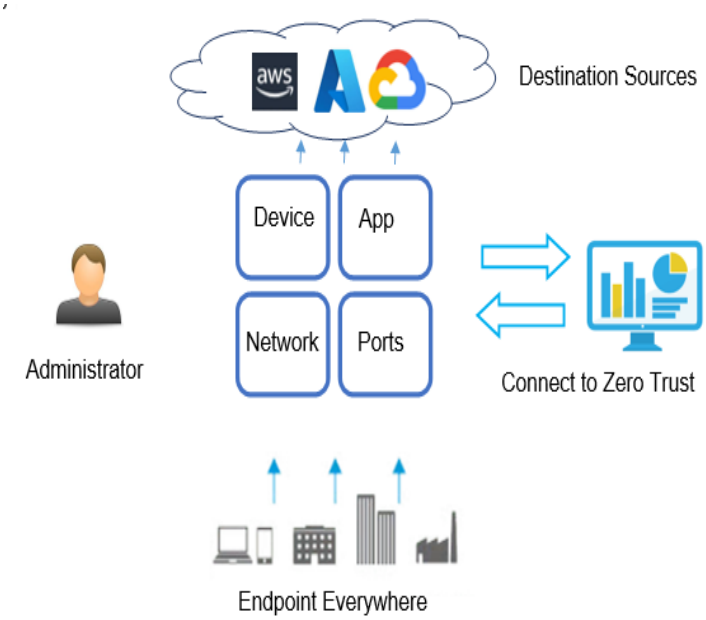

Fig 1: Zero trust Strategy Connector Architecture

IV. DESIGN ARCHITECTURE AND CONSIDERATIONS

Based on standard design assessment we have considered few designs requirement of the design, they are

a. DESIGN CONSIDERATION:

1. Compute
   Operating system: Windows /Linux
   Server: Microsoft Server 2016 Base Windows Image (for server endpoint)
   Computing service: AWS cloud services
   Python package: Python 2.0 and above
   Node package: Node JS 16.17.0 64 bit
2. Network
   Gateway and virtual Network: AWS VPC Service (Server Endpoint)
   Network Adapters: Standard Wi-Fi or Ethernet Adapter Enabled
3. Storage:
   Database: AWS DynamoDB, S3 Buckets

b. DESIGN SCOPE

The Solution established mainly to enable Zero trust security strategy and increase endpoint access defense and achieve network protection by securing traffic, data, and infrastructure. Endpoint systems include servers, end-user machine and remote mobile devices.

c. DESIGN STRAGTEGY

The conceptualize of software requirements with design challenges is addressed with take on design strategy like object-oriented approach with bottom-up design. The various subcomponent like network analysis, device health, endpoint status and other server component together into client server endpoint zero trust one component

d. DESIGN HIGHLIGHTS

The design feature includes the solution highlight like

1. Storage flexibility
2. Application Security
3. High availability
4. Remote accessing
5. Network insights

V. IMPLEMENTATION

a. IMPLEMENTATION OF CLIENT REQUEST HANDLER

To prevent the pre-event challenges the api request need to be protected in terms of vulnerability and scalability, so the request handler is always available to the endpoint.

All the pre-event api is authenticated with two factor authentication and redirected for post event after processing. The preprocessing rules consists of below validations

a. Check for device type in each metrics and update based on the availability to the database.
b. All api request need to analyze for request type and redirected to the respective post event as per the request type value.
c. Pre event method need to prevent exception without processing further in post event
d. Parameters related to the pre-event request need to be tracked and monitored

The below algorithm explains the flow of the prevent and post event in the API Gateway

Algorithm Start
Starting point: Request from API Gateway
Check for Token in header
Verify the Token
Check for Lambda type
if lambda present: publish to lambda or send error response
Check for request type in lambda
If request present: process request or send error response
If request type present: Get request type or send error message
Get response = Call () request type function
if Get Response === “Success”: Insertion Successful or Insertion error
Covert response into Json
Send Response
#Stop Algorithm

b. IMPLEMENTATION OF MULTI FACTOR AUTHENTICATION

The AWS Cognito enables a unique policy-based access to the user that is applied to all the resources and enables the strong authentication across the devices and their business,
The Configuration and policy followed in the AWS Cognito is as described below,

a. multifactor authentication

to increase the security with stronger authentication we have enabled the required user policy with multifactor sign-in and Sign-out, the user sign-in would automatically configure the required user on adaptive approach where the user can be registered with any third-party application.

b. Recover user information

Whenever a user forgot his password or user information the authentication service needs to support preferred way of recovery option to the user, here the account can be recovered either by phone number or through email.

c. Advanced Security

The advance security option can be enabled to the user policy to enable adaptive authentication. The most used authentication technique in this project is JsonWebToken (JWT) and OAuth for client server connection, and the token for each endpoint would be randomly updated on interval basis.

d. Application Integration

The application integration with the auth service is configured with custom domain and URL, the hosted URL provides OAuth 2.0 authentication and authorization with automatically built-in custom pages where the user can perform sign-in without any external application or console.

e. Identity providers

Here we have adopted the google identity provider to enable two factor authentication option like verification code and QR scanning for all the user in the application.

VI. CODING STANDARDS

PYTHON CODING STANDARDS

1. 4 space indentation and no tabs

The alignment of each function block and other conditional block, consists of 4-spaces spacing in start and end of block.

2. Use Comments

Each conditional block consists of single line comments and the functional blocks consists of description in multi-line comments that are represented in standard English lexicon

3. File path

Both absolute file and relative file path is affixed with ‘r’ character, to clearly mention the users about the file path and to address escape sequence related to special characters used in the path.

4. Use docstrings

Some characters need to be encoded with ‘‘single quotes and strings need to be encoded with ““double quotes.

5. Wrap lines with max 79 characters

Each line in the code is maintained with not to exceed more than 79 characters, for achieving readability, the code is factored with spacing, naming, and limited code size.

6. Naming Convention

Functional Name	Conventional Rules
Constants	The constants are declared with uppercase, and underscore are used for separation of words
Global Variable Name	Each global variable is attached with a gl and underscore prefix and declared with global keyword
Local Variable Name	Lowercase with camel case naming conventions is applied to all scope based local variables
Functions Name	Function names are declared with Uppercase and underscore for separation
Module Name	All module names are implemented by lowercase and bumpy case
Class Names	All class name is implemented with prefix as cl and underscore

Table 1: Naming Conventions Python

VII. CONCLUSIONS

It is insufficient to limit security measures to the usual ones, such as border protection, in-depth security methods, and physical security for endpoints and organisation networks. The workforce is dispersed across multiple locations, the workload is growing, and ongoing storage in external resources like the cloud makes it impossible to enforce a single security regulation on every endpoint within the organisation. Despite the organization's investment in mediator tools for endpoint security, the system became very compartmentalised and impeded communication and maintenance.

This research's primary goal is to solve the shortcomings of contemporary security and offer an integrated solution that replaces the present compartmentalised method with a principle-based, data-centric approach that includes practical identity solutions for each individual. The solution identified with new zero trust architecture achieved an end to perimeter-based security operations in a phased way by identifying the proper use case and business demands.

As a continual approach, the implementation included effort in improving the current constraints with two layered components like trusted client endpoint and trusted server endpoint, to increase current organization security posture through

a. Effective authentication across endpoints
b. Increased security insights on endpoints
c. Ensure the best user experience
d. Installing new backup approach, restoration for all endpoints
e. Educating on external activity with smart notification
f. Easy report snapshot.

Rather than concluding the solution as a destination, the zero-trust transition is taken as a journey that need to be constantly modeled and adopted to new security practices, The project plays a role with good understanding of security principles that would support the current form of organization, and the future goals aimed to improve successful implementation of security strategy, user experience and, impact endpoints with automated continuous verification and validation for all types of endpoints across real-time connected networks.

REFERENCES

1. *Casimer DeCusatis, Piradon Lientiraphan, Anthony sager, Mark pinelli Zero Trust Security Strategy: Implementing Zero Trust Cloud Networks with Transport Access Control and First Packet Authentication.*
2. *Zang Xiojian, Chen Liondong, Fan Jie, Wang Xiangqun, Zero Trust Security Strategy: Power Iot Security Protection Architecture based on zero trust framework.*
3. *Simone Rodigari, Donna O'Shea, Pat McCarthy, Martin McCarry, Sean McSweeney Zero Trust Security Strategy: Performance Analysis of Zero-Trust Multi-Cloud.*
4. *Saima Mehraj, M Tariq Bandey, Zero Trust Security Strategy: Establishing a Zero Trust Security Strategy in Cloud Computing Environment*
5. *Daniel D'Silva, Dayananda D. Ambawade, Zero Trust Security Strategy: Building a Zero Trust architecture using Kubernetes.*
6. *AWS documentation https://docs.aws.amazon.com/index.html.*
7. *https://docs.microsoft.com/en-us/security/zero-trust/zero-trust-overview -Microsoft Zero Trust Architecture.*

