Zero Trust Design Using Cloud Native Computing Foundation (CNCF) Landscape

Design Project

Agenda

1	Introduction	• Context / Situation.	6	Concepts of Operations	• Tools utilized in the project.
2	Importance of ZTA in Cybersecurity	• Explaining Zero Trust (ZT) principles • Introduce CNCF security tools	7	Methodology and Research Design	• Parallel investigation of Azure and local VMs. • Major research points.
3	Stakeholders	• List of all individuals, groups, or organizations that have an interest or concern in the project.	8	Implementation	• Demo & visuals showcasing ZTA aspects in both the paper and the presentation.
4	Problem Statement	• Brief description of the challenge that needs to be solved	9	Verification & Validation	• Ensure that the research and is connected to ZT principles. • Demonstrate how work ties back to ZT principles
5	Requirements	• The necessary components and expertise needed to implement the solution.	10	Application and Lessons Learned	• How organizations can utilize the research findings • Lessons learned to transition towards a ZTA approach.

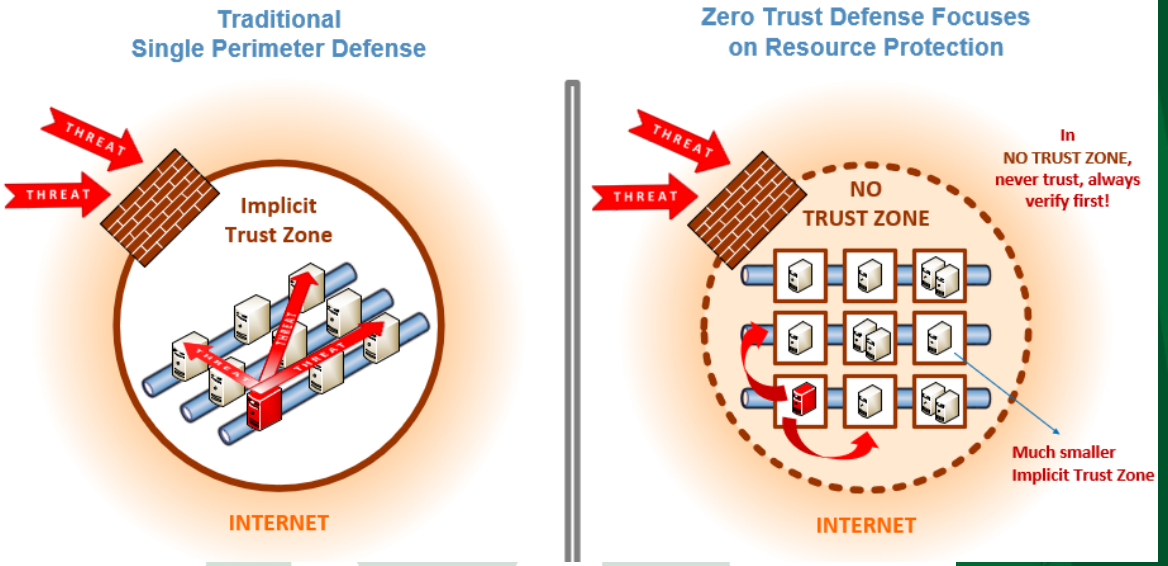
Introduction

Rising Need for Cybersecurity in Cloud-Native Environment

- Traditional perimeter-based security model inadequate for modern threats
- Zero Trust Architecture (ZTA) crucial amid cloud service proliferation

Example: Microsoft Exchange Server Vulnerabilities (2021)

- Microsoft Exchange Server vulnerabilities led to unauthorized email access and potential malware installation
- Highlighting the need for ZTA principles like continuous verification and least privilege access to prevent such breaches



https://rb.gy/y14krb

ZTA Integration

Integration of ZTA with CNCF Landscape

- CNCF offers open source security tools
- Research focuses on merging ZTA principles with CNCF tools
- Utilization of Azure platform and CNCF tools (Envoy, Vault, SPIFFE/SPIRE) in simulated cloud environment
- Demonstrates efficacy in securing cloud service communications through access controls and validation mechanisms (like nmap).

Importance of ZTA in Cybersecurity

Trust/Verify Paradigm

ZTA emphasizes the principle of "Never trust, always verify,"[1] ensuring continuous authentication of access requests regardless of user location or network environment.

Multi-Factor Authentication & Encryption

ZTA incorporates robust security measures such as MFA and encryption to protect critical data, reducing the risk of unauthorized access and data breaches.

Micro-segmentation

ZTA employs micro-segmentation to minimize the attack surface, preventing lateral movement of potential attackers within the network and enhancing overall security posture.

Identity and Access Management (IDAM)

ZTA shifts from perimeter-based defenses to granular access controls, enhancing security by closely managing user identities and permissions.

Problem Statement

Proprietary Zero Trust (ZT) products like **Zscaler** and **Okta** are too expensive for many organizations. However, open-source alternatives within the Cloud Native Computing Foundation (CNCF) offer robust cybersecurity solutions without hefty costs. By embracing CNCF's open-source offerings, we can make effective cybersecurity accessible to organizations of all sizes and budgets.

Stakeholders

Users and administrators of cloud-native applications

Decision-makers responsible for cybersecurity strategy within organizations

Cloud service providers like Azure

Entities concerned with regulatory compliance and risk management

MITRE, the company sponsoring the project

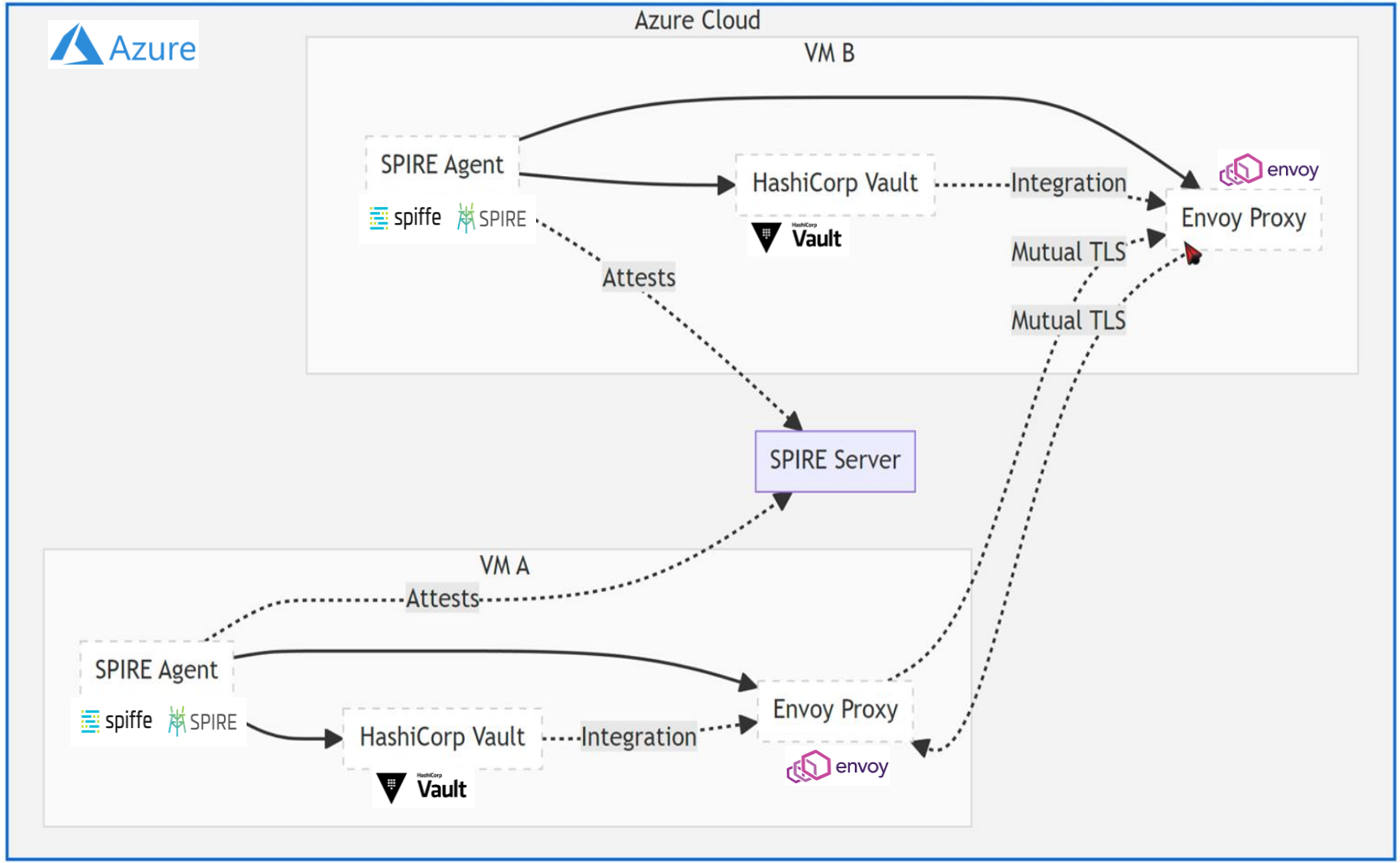
The mentors and resources needed for the project, from George Mason University (GMU)

Requirements

To effectively implement the project, the following requirements must be met:

1) **Comprehensive Understanding of ZTA Principles**
2) **Exploration and Research of CNCF Tools**
3) **Effective Implementation and Verification of CNCF Tools**
4) **Relating Results to ZTA Principles**
5) **Dissemination of Research Findings**

By adhering to these requirements, the project aims to contribute significantly to the understanding and implementation of ZTA principles and CNCF tools within the industry.


Azure
Azure Cloud
VM B
SPIRE Agent
spiffe
SPIRE
HashiCorp Vault
Vault
Integration
envoy
Envoy Proxy
Mutual TLS
Attests
Mutual TLS
SPIRE Server
VM A
Attests
SPIRE Agent
spiffe
SPIRE
HashiCorp Vault
Vault
Integration
Envoy Proxy
envoy

Methodology & Research Design

Methodology

Research and Analysis:

- **Studied Zero Trust principles** and adapted them to the cloud environment.
- Conducted a systematic **analysis of open-source technologies available** within the CNCF Landscape to identify suitable tools for implementing Zero Trust Architecture (ZTA).

Architecture Design, Prototype Development, and Deployment:

- Designed a custom Zero Trust Architecture tailored to the cloud environment, specifying the **integration of SPIFFE_SPIRE, Vault, Envoy, and visualized data through the use of Splunk.**
- Developed and deployed a prototype ZTA system into the production cloud environment, integrating ZTA components with existing cloud infrastructure and services.

Testing, Validation, and Evaluation:

- Performed **security assessments using tools like Nmap** to identify vulnerabilities and ensure compliance with security policies.
- Evaluated the performance and security of the implemented ZTA, iterating on the design and implementation to address any identified issues or enhancements specific to the cloud environment.

Documentation:

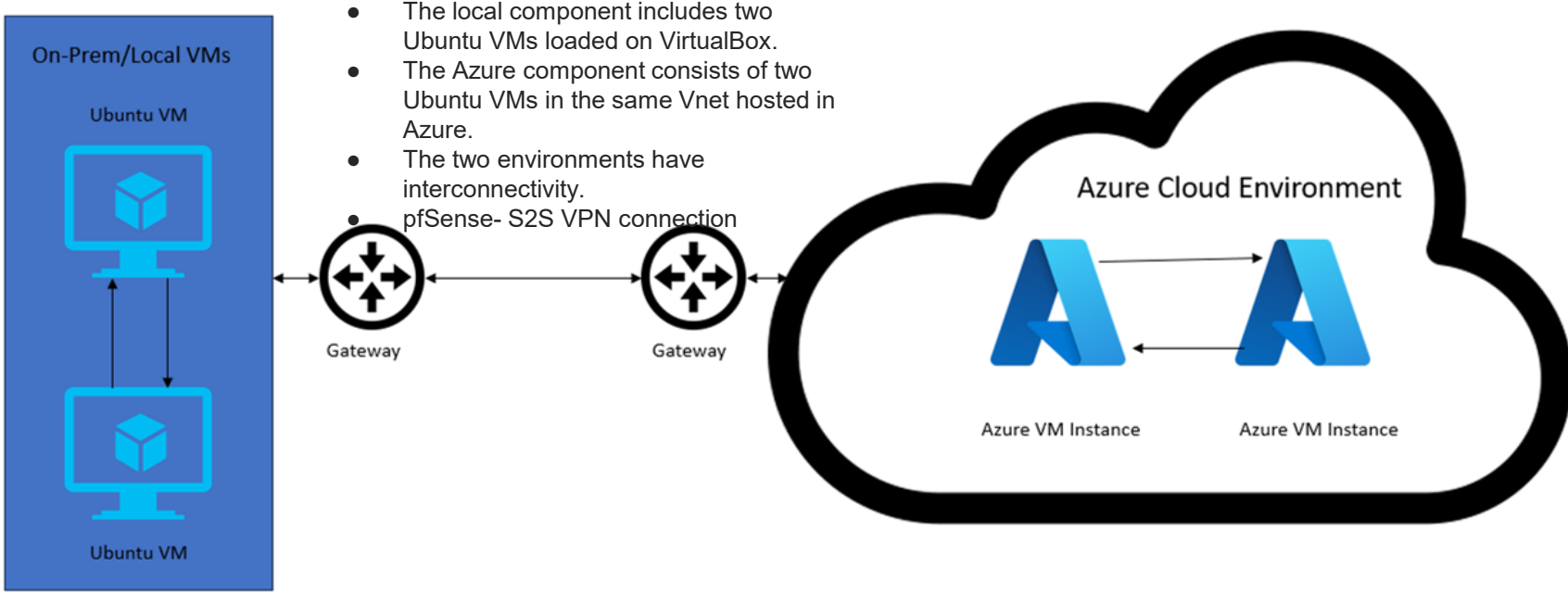
- Documented the ZTA architecture, configurations, and integration points specific to the cloud environment to ensure comprehensive understanding and maintainability of the system.

Initial Environment Set-up

Research Design

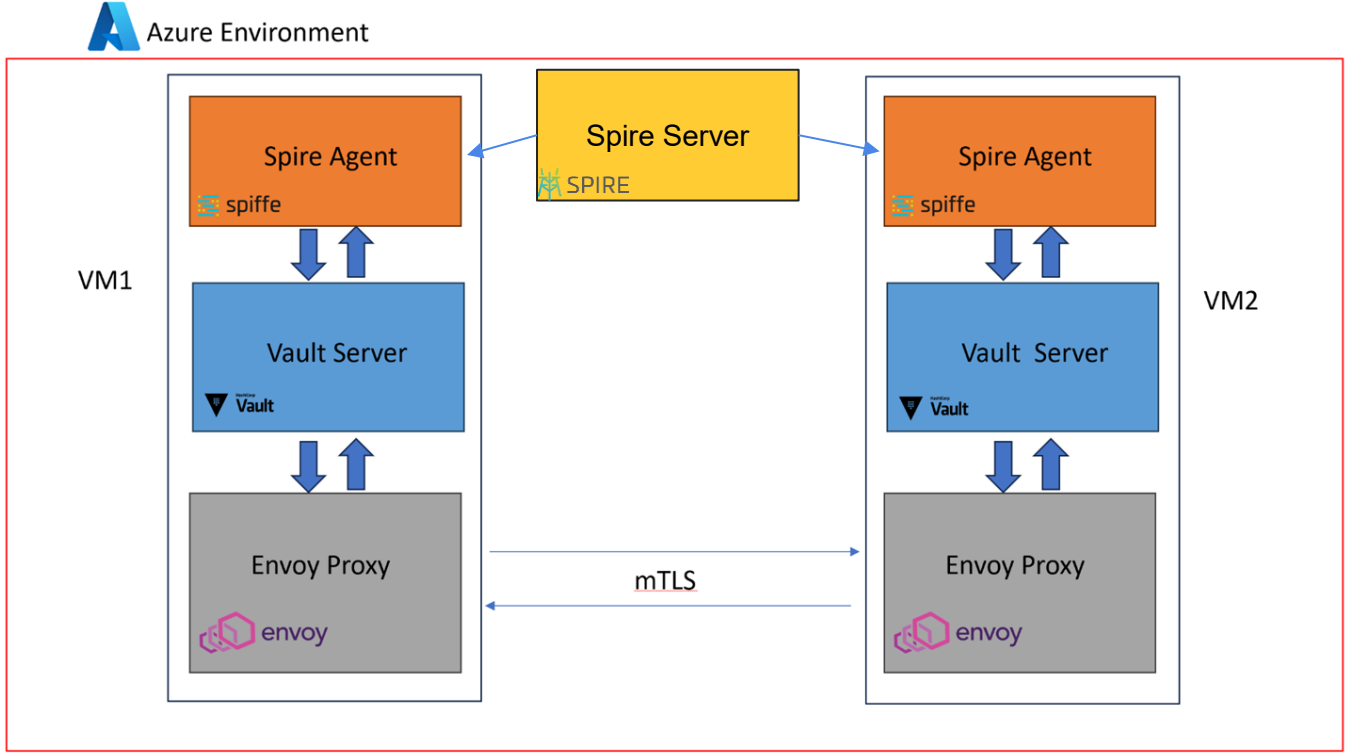
A Hybrid Cloud environment hosted locally and in Azure.

- The local component includes two Ubuntu VMs loaded on VirtualBox.
- The Azure component consists of two Ubuntu VMs in the same Vnet hosted in Azure.
- The two environments have interconnectivity.
- pfSense- S2S VPN connection



Final Environment Set-up

Research Design

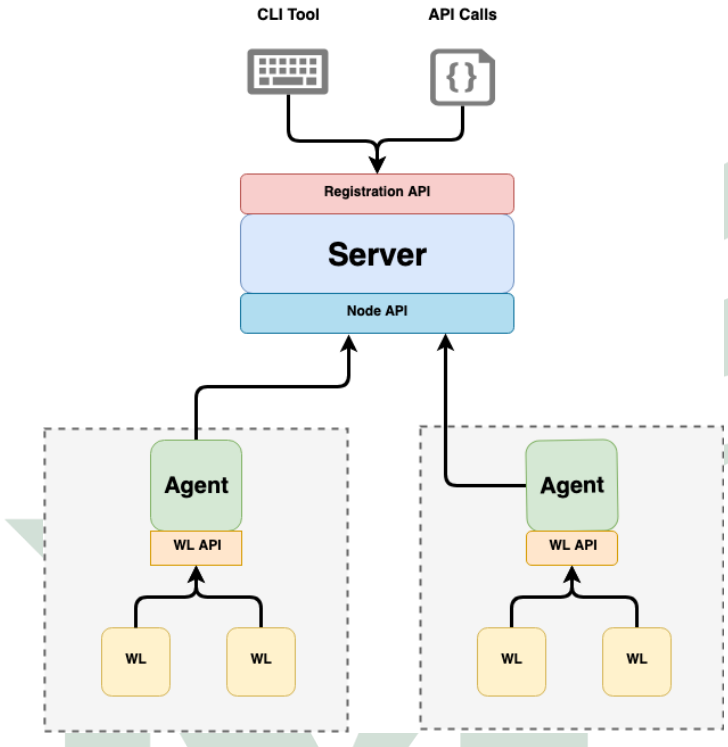


Implementation

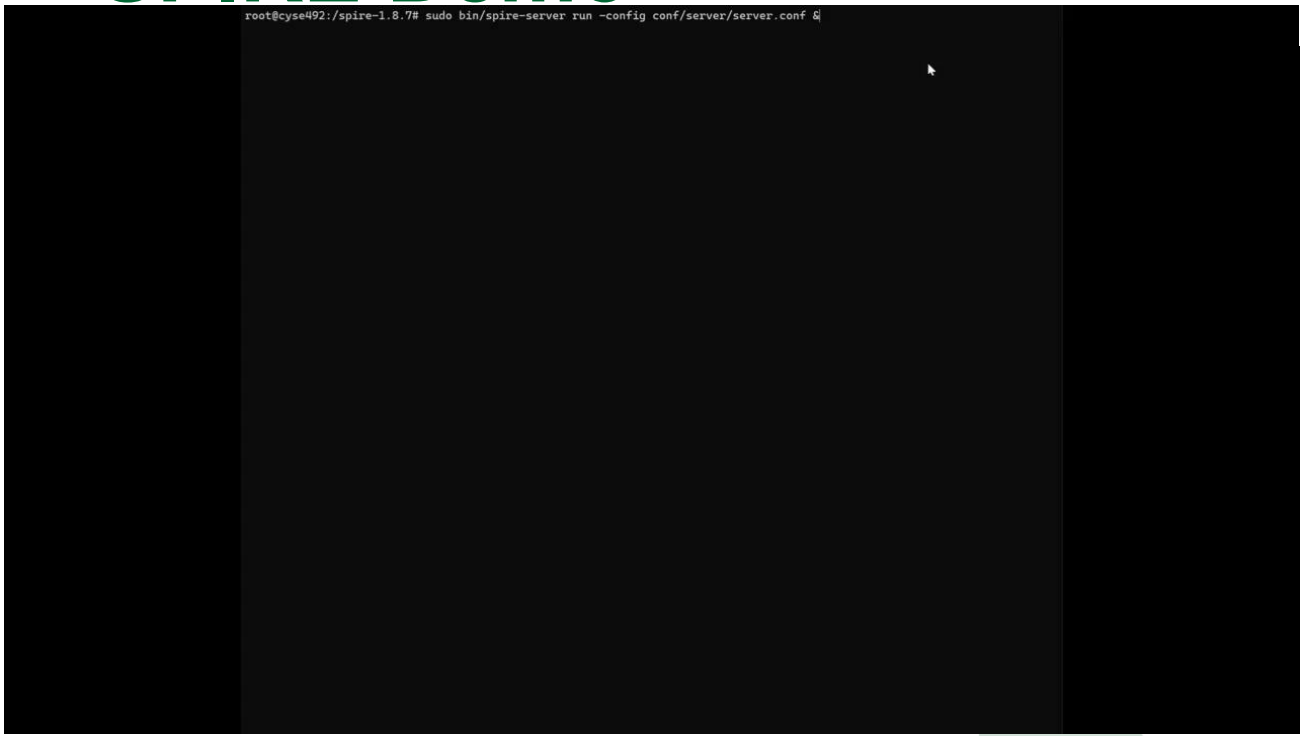
Technical Depth and Demonstration

SPIRE

- Security application consisting of a SPIRE server and agent used to register workloads and retrieve SVIDs.
- Uses SVIDs for secure authentication between cloud services.
- Unique verifiable SPIFFE IDs are created for services registered to the SPIRE server.
- Typical deployments consist of a SPIRE server and multiple SPIRE agents.
- The SPIRE Server:
  - Generates and distributes the SPIFFE IDs.
  - Registers workloads.
  - Attests nodes.
  - Signs keys.
- SPIRE Agents:
  - Retrieve SPIFFE IDs and store them until they are needed by a workload.
  - Attest the identity of a workload.
  - Supply the workload with its SPIFFE ID.



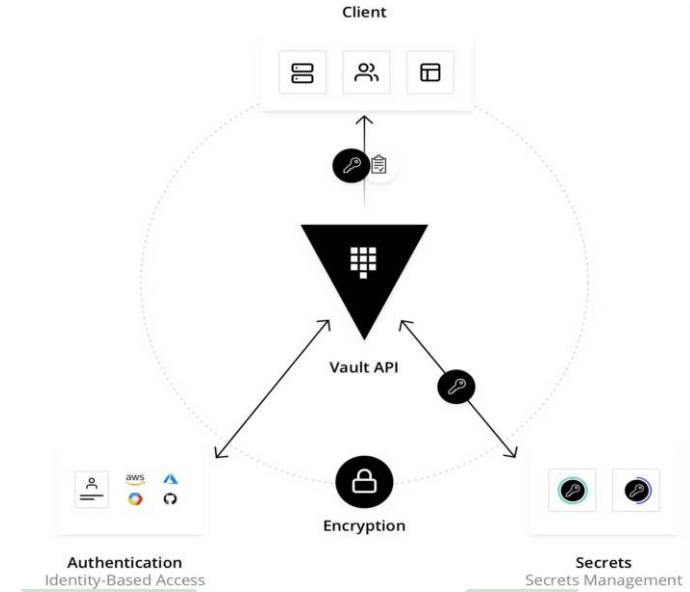
SPIRE Demo



Link: SPIRE Demo

Vault

- Used by applications that need to transmit data securely to store the keys and certificates.
- Comprehensive secrets management solution.
- Centralizes the storage, issuance, management of sensitive credentials such as:
  - Tokens
  - Passwords
  - Certificates
  - Encryption keys
- Incorporates tight access controls, sophisticated leasing and renewal mechanisms, and detailed audit logging.
- Vault's API-driven approach allows seamless integration with other services and applications, thereby enhancing automation and scalability in a cloud ecosystem

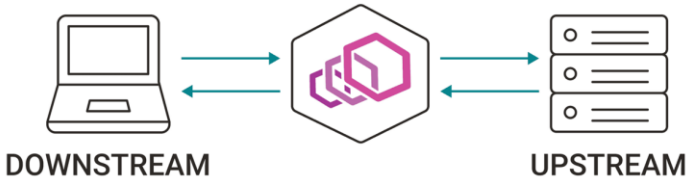


Vault Demo

Link: [Vault Demo]

Envoy

- Serves as a high-performance, cloud-native proxy that plays a crucial role in implementing mutual TLS (mTLS) for secure service-to-service communication within a Zero Trust architecture.
- Has sophisticated service mesh capabilities that allow it to forge verified communication tunnels, leveraging the SPIFFE IDs from SPIRE to ensuring authentication and encryption.
- Most deployments use it as a service mesh proxy that facilitates service to service communications securely using mTLS



codilime

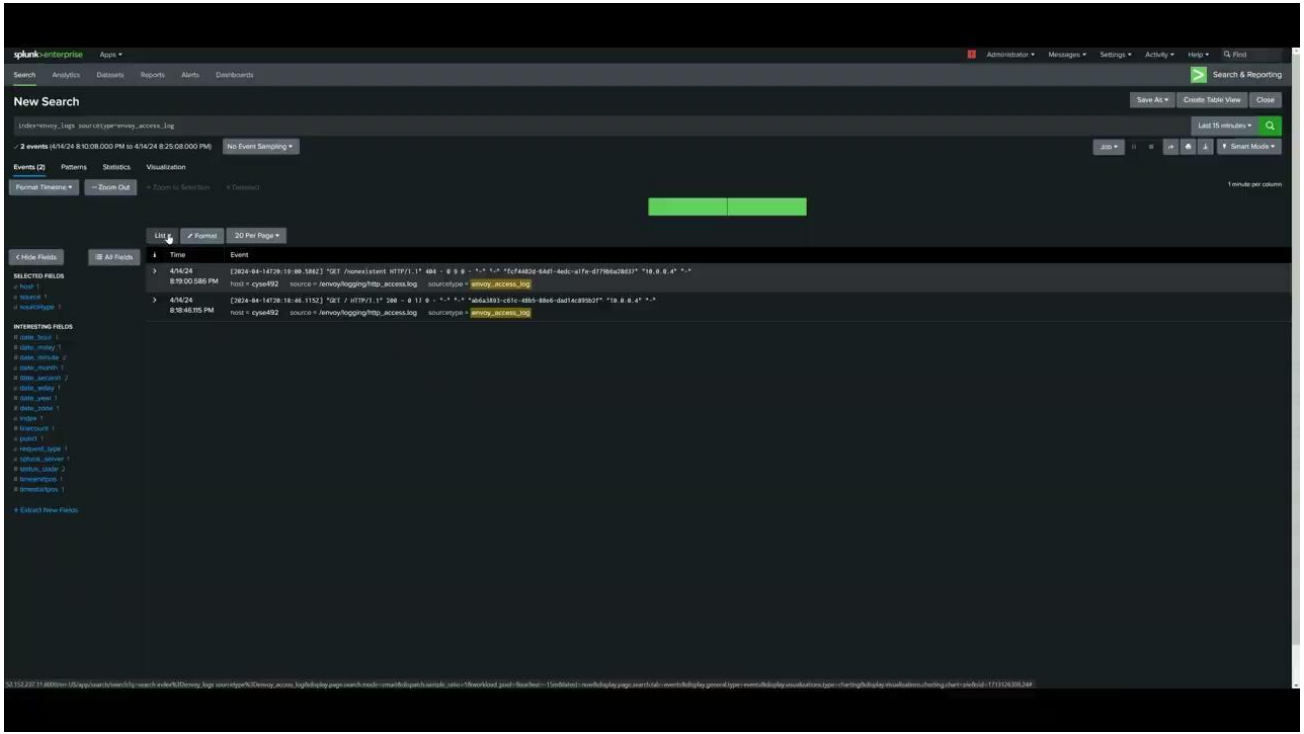
Envoy Demo

Link: Envoy Demo

Visualizing Our Data

We Used:

splunk>



CNCF Alternatives:

fluentd

OpenSearch

Link: [Splunk Demo]

Verification & Validation

We verified and validated our solutions by conducting tests and experiments with **Nmap**. This included tests for the functionality of CNCF tools in enforcing Zero Trust principles, evaluating the effectiveness of access controls and validation mechanisms, and ensuring compliance with security standards and best practices.

Additionally, we solicited feedback from stakeholders (MITRE & mentor professor) and incorporated their input to refine our solutions further.

Nmap

Why Nmap?

- Nmap helps scanning cloud VMs on Azure due to its versatility, open-source nature, and cross-platform compatibility.

- It offers comprehensive scanning techniques, scripting capabilities, and potential integration with Azure for network security assessments. However, adherence to Azure's security policies is crucial when using Nmap in Azure environments.

https://shorturl.at/ginX5

Nmap

Checking for user availability

```
group2@cyse492machine2:~$ ps -aux
USER         PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root           1  2.0  0.1 169608 12876 ?        Ss   21:25   0:02 /sbin/init
root           2  0.0  0.0      0     0 ?        S    21:25   0:00 [kthreadd]
root           3  0.0  0.0      0     0 ?        I<   21:25   0:00 [rcu_gp]
root           4  0.0  0.0      0     0 ?        I<   21:25   0:00 [rcu_par_gp]
root           5  0.0  0.0      0     0 ?        I<   21:25   0:00 [slub_flushwq
root           6  0.0  0.0      0     0 ?        I<   21:25   0:00 [netns]
root           7  0.0  0.0      0     0 ?        I    21:25   0:00 [kworker/0:0-
root           8  0.0  0.0      0     0 ?        I<   21:25   0:00 [kworker/0:0H
root           9  0.1  0.0      0     0 ?        I    21:25   0:00 [kworker/u4:0
root          10  0.0  0.0      0     0 ?        I<   21:25   0:00 [mm_percpu_wq
root          11  0.0  0.0      0     0 ?        S    21:25   0:00 [rcu_tasks_ru
root          12  0.0  0.0      0     0 ?        S    21:25   0:00 [rcu_tasks_tr
root          13  0.0  0.0      0     0 ?        S    21:25   0:00 [ksoftirqd/0]
root          14  0.0  0.0      0     0 ?        I    21:25   0:00 [rcu_sched]
root          15  0.0  0.0      0     0 ?        S    21:25   0:00 [migration/0]
root          16  0.0  0.0      0     0 ?        I    21:25   0:00 [kworker/0:1-
root          17  0.0  0.0      0     0 ?        S    21:25   0:00 [cpuhp/0]
root          18  0.0  0.0      0     0 ?        S    21:25   0:00 [cpuhp/1]
root          19  0.3  0.0      0     0 ?        S    21:25   0:00 [migration/1]
root          20  0.0  0.0      0     0 ?        S    21:25   0:00 [ksoftirqd/1]
root          21  0.0  0.0      0     0 ?        I    21:25   0:00 [kworker/1:0-
```

Nmap Scans

TCP SYN "-sS nmap IP"

TCP connect "-sT nmap IP"

Version detection "-sV nmap IP"

UDP "-sU nmap IP"

```
group2@cyse492:~$  nmap -sV 52.152.237.11
Starting Nmap 7.80 ( https://nmap.org ) at 2024-04-14 21:13 UTC
Note: Host seems down. If it is really up, but blocking our ping probes, try -Pn
Nmap done: 1 IP address (0 hosts up) scanned in 3.19 seconds
group2@cyse492:~$ nmap -Pn -sV 52.152.237.11
Starting Nmap 7.80 ( https://nmap.org ) at 2024-04-14 21:13 UTC
Nmap scan report for 52.152.237.11
Host is up (0.00089s latency).
Not shown: 995 filtered ports
PORT      STATE  SERVICE          VERSION
22/tcp    open   ssh              OpenSSH 8.2p1 Ubuntu 4ubuntu0.11 (Ubuntu Linux; protocol 2.0)
8000/tcp  open   http             Splunkd httpd
8081/tcp  closed blackice-icecap
8200/tcp  closed trivnet1
10000/tcp closed snet-sensor-mgmt
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 15.24 seconds
```

Nmap Penetration Testing


```
group2@cyse492machine2:~$ sudo nmap 173.66.84.98  -Pn -sV -O -p1-9999
Starting Nmap 7.80 ( https://nmap.org ) at 2024-04-03 13:32 UTC
Stats: 0:01:48 elapsed; 0 hosts completed (1 up), 1 undergoing SYN Stealth Scan
SYN Stealth Scan Timing: About 5.35% done; ETC: 14:05 (0:31:50 remaining)
Stats: 0:16:35 elapsed; 0 hosts completed (1 up), 1 undergoing SYN Stealth Scan
SYN Stealth Scan Timing: About 49.65% done; ETC: 14:05 (0:16:49 remaining)
Debugging Increased to 1.
Stats: 0:31:55 elapsed; 0 hosts completed (1 up), 1 undergoing SYN Stealth Scan
SYN Stealth Scan Timing: About 95.61% done; ETC: 14:05 (0:01:28 remaining)
Current sending rates: 9.99 packets / s, 352.08 bytes / s.
Overall sending rates: 9.98 packets / s, 439.24 bytes / s.
Packet capture filter (device eth0): dst host 10.0.0.5 and (icmp or (tcp and (src host 173.66.84.98)))
NSE: Script scanning 173.66.84.98.
NSE: Starting runlevel 1 (of 2) scan.
NSE: Starting runlevel 2 (of 2) scan.
Nmap scan report for pool-173-66-84-98.washdc.fios.verizon.net (173.66.84.98)
Host is up.
All 9999 scanned ports on pool-173-66-84-98.washdc.fios.verizon.net (173.66.84.98) are filtered
Too many fingerprints match this host to give specific OS details
TCP/IP fingerprint:
SCAN(V=7.80%E=4%D=4/3%OT=%CT=%CU=%PV=N%G=N%TM=660D6238%P=x86_64-pc-linux-gnu)
U1(R=N)
IE(R=N)


Read from /usr/bin/../share/nmap: nmap-os-db nmap-payloads nmap-service-probes nmap-services.
OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 2013.34 seconds
group2@cyse492machine2:~$
```

Penetration Testing


```
          (__)    )\
            ||--|| *


       =[ metasploit v6.4.1-dev-                          ]
+ -- --=[ 2404 exploits - 1239 auxiliary - 422 post       ]
+ -- --=[ 1465 payloads - 47 encoders - 11 nops           ]
+ -- --=[ 9 evasion                                       ]

Metasploit Documentation: https://docs.metasploit.com/

msf6 > search Shellshock

Matching Modules
================

   #   Name                                              Disclosure Date  Rank       Check  Description
   -   ----                                              ---------------  ----       -----  -----------
   0   exploit/linux/http/advantech_switch_bash_env_exec 2015-12-01       excellent  Yes    Advantech Switch Bash Environment Variable Code Injection (Shellshock)
   1   exploit/multi/http/apache_mod_cgi_bash_env_exec   2014-09-24       excellent  Yes    Apache mod_cgi Bash Environment Variable Code Injection (Shellshock)
   2     \_ target: Linux x86                            .                .          .      .
   3     \_ target: Linux x86_64                         .                .          .      .
   4   auxiliary/scanner/http/apache_mod_cgi_bash_env    2014-09-24       normal     Yes    Apache mod_cgi Bash Environment Variable Injection (Shellshock) Scanner
   5   exploit/multi/http/cups_bash_env_exec             2014-09-24       excellent  Yes    CUPS Filter Bash Environment Variable Code Injection (Shellshock)
   6   auxiliary/server/dhclient_bash_env                2014-09-24       normal     No     DHCP Client Bash Environment Variable Code Injection (Shellshock)
   7   exploit/unix/dhcp/bash_environment                2014-09-24       excellent  No     Dhclient Bash Environment Variable Injection (Shellshock)
   8   exploit/linux/http/ipfire_bashbug_exec            2014-09-29       excellent  Yes    IPFire Bash Environment Variable Injection (Shellshock)
   9   exploit/multi/misc/legend_bot_exec                2015-04-27       excellent  Yes    Legend Perl IRC Bot Remote Code Execution
   10  exploit/osx/local/vmware_bash_function_root       2014-09-24       normal     Yes    OS X VMWare Fusion Privilege Escalation via Bash Environment Code Injection (Shellshock)
   11  exploit/multi/ftp/pureftpd_bash_env_exec          2014-09-24       excellent  Yes    Pure-FTPd External Authentication Bash Environment Variable Code Injection (Shellshock)
   12    \_ target: Linux x86                            .                .          .      .
   13    \_ target: Linux x86_64                         .                .          .      .
   14  exploit/unix/smtp/qmail_bash_env_exec             2014-09-24       normal     No     Qmail SMTP Bash Environment Variable Injection (Shellshock)
   15  exploit/multi/misc/xdh_x_exec                     2015-12-04       excellent  Yes    Xdh / LinuxNet Perlbot / fBot IRC Bot Remote Code Execution


Interact with a module by name or index. For example info 15, use 15 or use exploit/multi/misc/xdh_x_exec
```

Connection & Relevance to ZTA Principles

- The Cloud-Based VM Setup within the Azure Environment enforces **the principle of least privilege**, ensuring no inherent trust for entities inside or outside the network perimeter.
- Integration with SPIFFE_SPIRE, Vault, and Envoy facilitates **robust authentication mechanisms** and secure communication channels.
- SPIFFE_SPIRE generates and manages SPIFFE IDs and X.509 SVID certificates, ensuring **strict identity verification for services**.
- Vault serves as a centralized repository for managing secrets and **providing access control**.
- Envoy enforces access controls and establishes mutual TLS connections for **secure communication.**
- Nmap complements the architecture by conducting security assessments, aligning with ZTA's principle of **continuous monitoring.**

Overall, our architecture embraces the core tenets of Zero Trust, **enhancing security posture and mitigating risks** in our cloud environment.

Application and Lessons Learned

– Conclusion

- Embracing Zero Trust principles, which advocate skepticism towards every access request regardless of user location, offers organizations a powerful approach to enhancing their security posture amidst evolving cyber threats.

- Integration of CNCF tools like Envoy and SPIFFE/SPIRE within a simulated Azure cloud environment showcases the effectiveness of stringent access controls and continuous validation mechanisms.

- This research contributes not only to advancing understanding and implementation of Zero Trust Architecture (ZTA) but also establishes a robust foundation for future exploration and development in the dynamic field of cloud security.

CONOP

Azure Environment:
- Orchestrates the deployment of necessary resources for system operation.

Cloud-Based VM Setup:
- Encompasses machine1 and machine2.
- Integrates with Envoy, SPIFFE_SPIRE, and Vault.
- Facilitates secure communication and management of identities and secrets.

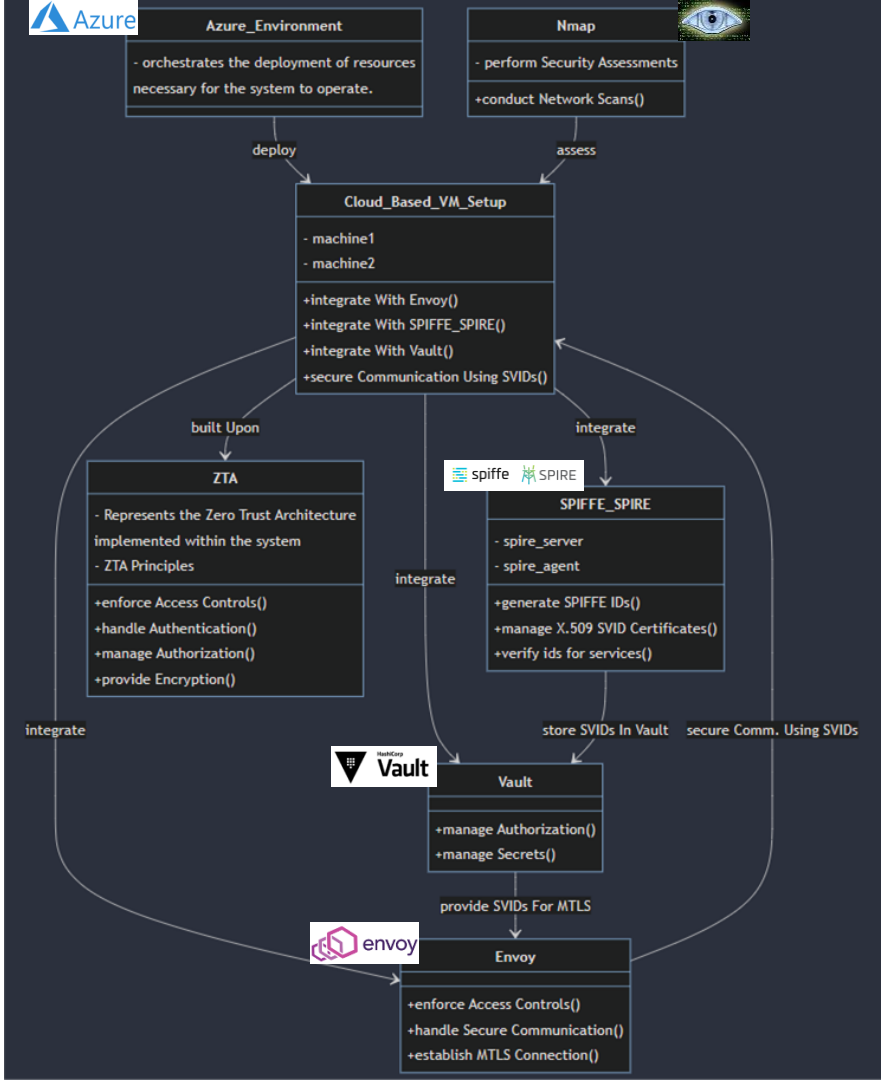
SPIFFE_SPIRE:
- Includes a SPIRE server and SPIRE agent.
- Generates SPIFFE IDs and manages X.509 SVID certificates.
- Verifies identities for services.

Vault:
- Manages authorization and secrets securely.

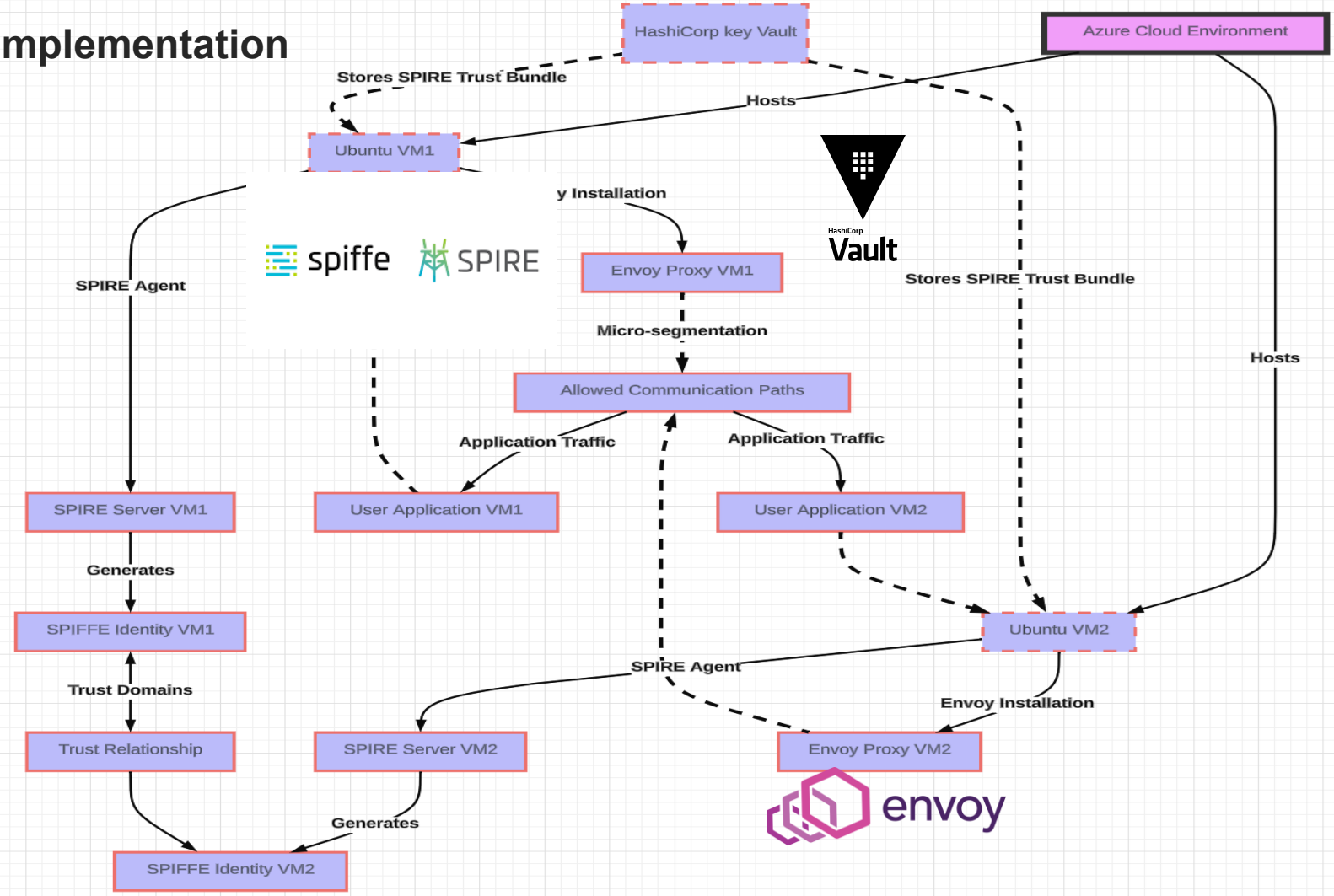
Envoy:
- Enforces access controls.
- Handles secure communication.
- Establishes mutual TLS connections.

Nmap:
- Performs security assessments.
- Conducts network scans to identify vulnerabilities.

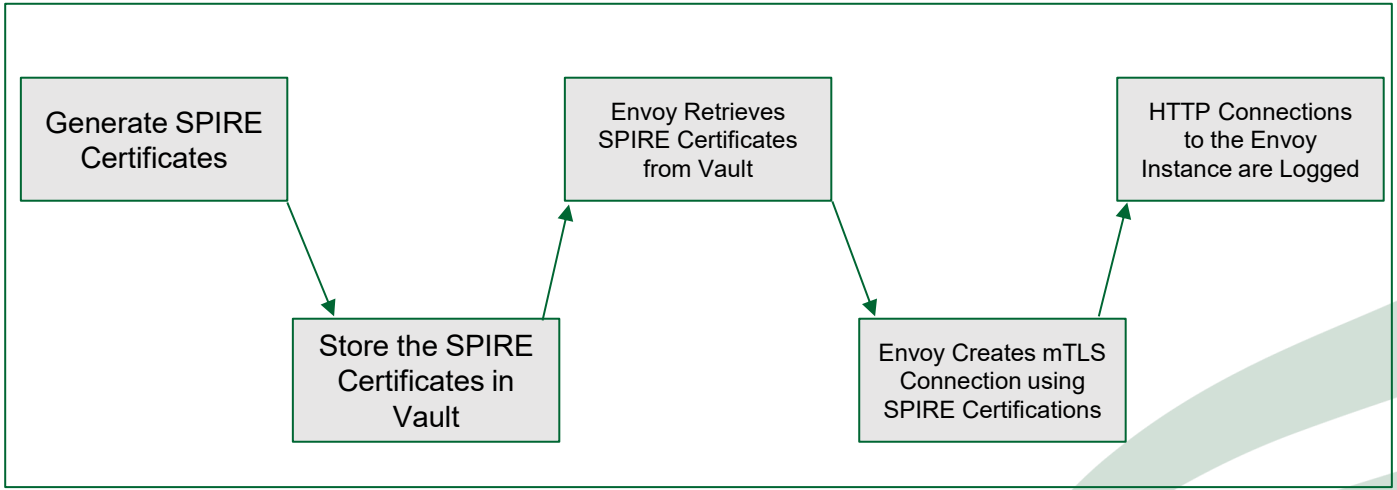
ZTA (Zero Trust Architecture):
- Represents the architectural principle.
- Enforces access controls, authentication, authorization, and encryption.



Implementation



Data Model:


Generate SPIRE Certificates
Store the SPIRE Certificates in Vault
Envoy Retrieves SPIRE Certificates from Vault
Envoy Creates mTLS Connection using SPIRE Certifications
HTTP Connections to the Envoy Instance are Logged

Why Azure?

Research Design

- **Seamless Integration:** Azure integrates smoothly with other Microsoft Azure services, ensuring compatibility and simplifying implementation.
- **Robust Identity Management:** Offers centralized user authentication, authorization, and access control, enabling granular access controls across cloud-native applications.
- **Scalability and Flexibility:** Highly scalable to accommodate future growth and supports both cloud-only and hybrid identity scenarios.
- **Enhanced Security Features:** Incorporates advanced security features like multi-factor authentication, conditional access policies, and identity protection capabilities.
- **Comprehensive Monitoring and Reporting:** Provides insights into user activities, security events, and compliance status, facilitating proactive threat detection and compliance reporting.

https://shorturl.at/ALM27

Azure

https://shorturl.at/jxKPQ

Researched CNCF Tools

kubernetes

HELM

PROJECT CALICO

Falco

Istio

Open Policy Agent

Network Service Mesh

HARBOR

LINKERD

SPIRE

envoy

HashiCorp Vault

How the Tools Can Be Used Pt.1

- **Kubernetes** comes with Role-Based-Access-Control capabilities. Network policy implementation can determine how pods communicate with each other and outside services. IAM systems can also be integrated to consistently enforce authentication within clusters.
- **Helm** enables audit logging in the Kubernetes environment and also has rollback features to revert dacomes with Role-Based-Access-Control capabilities. Network policy implementation can determine how pods communicate with each other and outside services. IAM systems can also be integrated to consistently enforce authentication within clusters.
- mages. There are many implementable features which can run vulnerability scans and enable access control for users.
- **Falco** is a tool that alerts on security threats, unusual behavior, and compliance violations. These features can be used to generate alerts if any Zero-Trust policies are compromised or violated.
- **Harbor** is able to use Role-Based-Access-Control, check for vulnerabilities within an image, and sign trusted images and is designed to work with major cloud tools such as Kubernetes and Docker. This can be used to implement ZTA by implementing access control and ensuring all images are safe and signed as trusted.

Juny

How the Tools Can Be Used Pt.2

- **Linkerd** is able to be used in ZTA because of its zero configuration mutual TLS and Zero-Trust Policy feature. This feature sets authorization policies that determine what type of traffic is able to reach meshed Kubernetes pods.
- **Network Service Mesh** is a Hybrid/Multi-cloud IP Service Mesh and one of its key features is Layer 3 Zero Trust. This service can allow different workloads and databases interact with each other while maintaining Zero Trust.
- **Open Policy Agent** provides a unified framework and approach to applying policies across cloud environments. Since a big part of ZTA is policy management, OPA assists this process by providing a standardized approach.
- **Istio** is a service mesh that provides Traffic Management, Observability, and Security to Kubernetes clusters. In terms of security, Istio provides Role-Based-Access Control and authentication, two important aspects of ZTA.
- **SPIFFE and SPIRE** provide identity management for cloud infrastructure. One of the key use cases of SPIFFE and SPIRE is cross service authentication for ZTA.

Ryan

Early Steps - Local VM Development

Emphasizing Azure VM Development for ZTA Architecture

- During the early stages, the team divided their focus into **two environments**. Due to time restraints, the team decided to stop research on the **local VM** end and to completely shift the focus onto developing t**he Azure VMs**. This was in the best interest for team and the overall story of implementing Cloud Native tools to generate the ZTA architecture as defined by the project guidelines.

- The local VM team focused on incorporating CNCF tools such as **Kubernetes**, **Calico**, and **Helm** to create a secure environment which would later be connected to the Azure VMs through a **S2S VPN connection(pFsense)**. From there, the team would be able to establish secure communications and demonstrate the overarching ZTA architecture over both environments.

Technical Approach

Local VM Environment Setup:

- **Local VM Project:**
  - Create a new project on Oracle VirtualBox
- **Local VM Configuration:**
  - Define the specifications of the VM
  - Download Ubuntu 22.04
- **Tool Installation:**
  - Install necessary CNCF tools to create a ZTA environment
- **Network Segmentation and User Role Specification:**
  - Install a tool to help ensure network isolation
  - Declare roles based on users using tools such as Kubernetes
- **Network Traffic**
  - Implement tools to filter and/or monitor network traffic (Calico, Helm)
  - Ensure secure communication

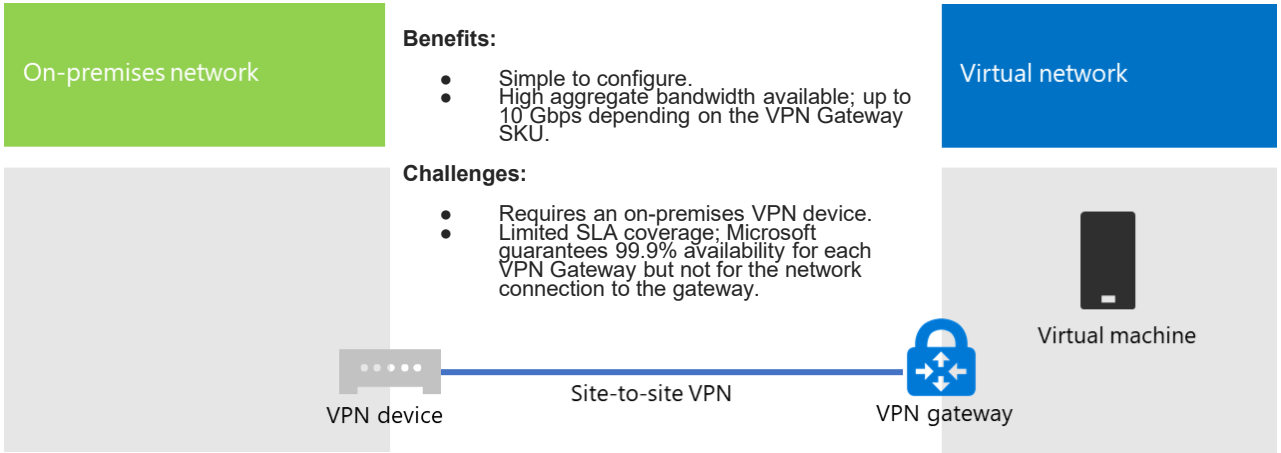
Testing and Experimentation:

- **Test Environments:**
  - Create multiple test environments to simulate different scenarios.
- **Testing Tools**
  - Penetration testing to test security of the ZTA structure (Katoolin)
- **Scaling:**
  - Test scalability of applications and infrastructure.
- **Documentation:**
  - Document installation process and issues encountered

VPN Connection

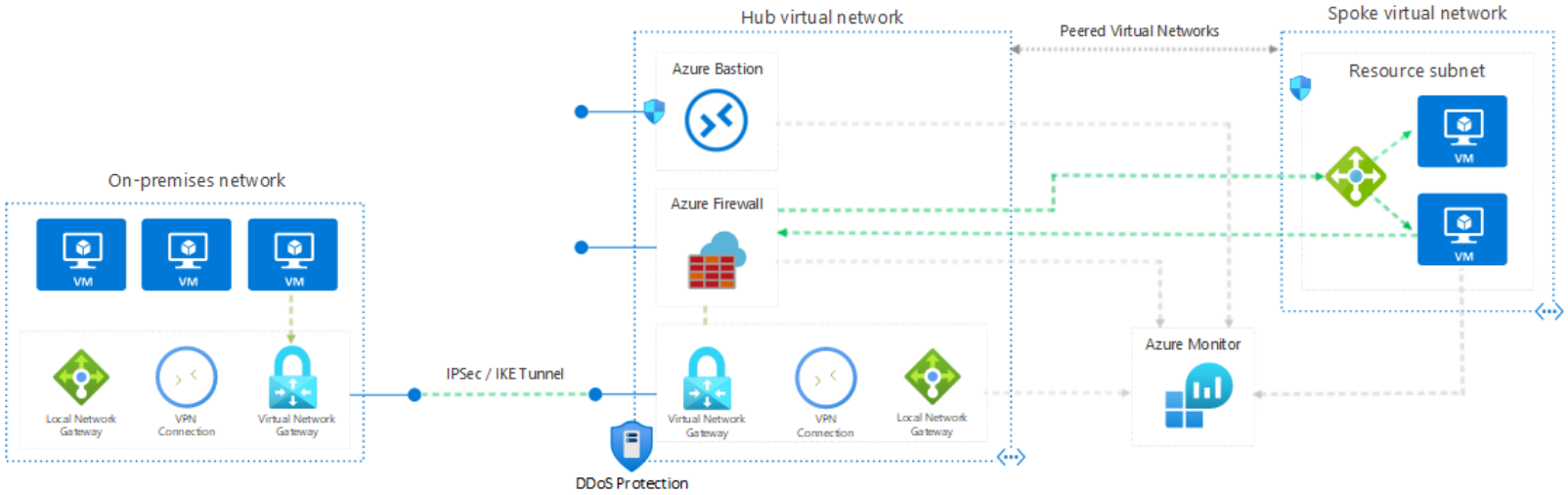
Option1

Description: A VPN connection establishes an encrypted tunnel between an Azure virtual network and an on-premises location, facilitating secure communication over the public internet.



[Connect an on-premises network to a Microsoft Azure virtual network - Microsoft 365 Enterprise | Microsoft Learn]

Example of secure hybrid network

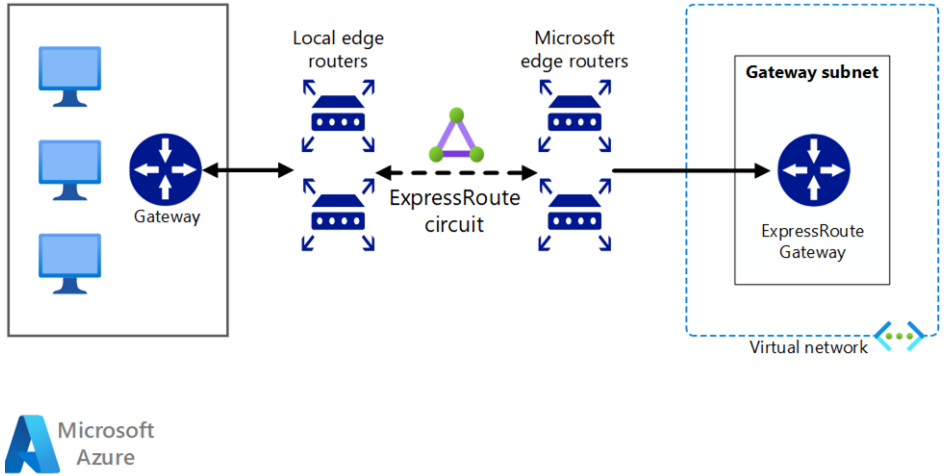


[Implement a secure hybrid network - Azure Architecture Center | Microsoft Learn]

Azure ExpressRoute

Option 2

Description: Azure ExpressRoute provides a private, dedicated connection through a third-party connectivity provider, offering high bandwidth and lower latencies compared to typical internet connections.



On-premises network using ExpressRoute - Azure Architecture Center | Microsoft Learn

Challenges:

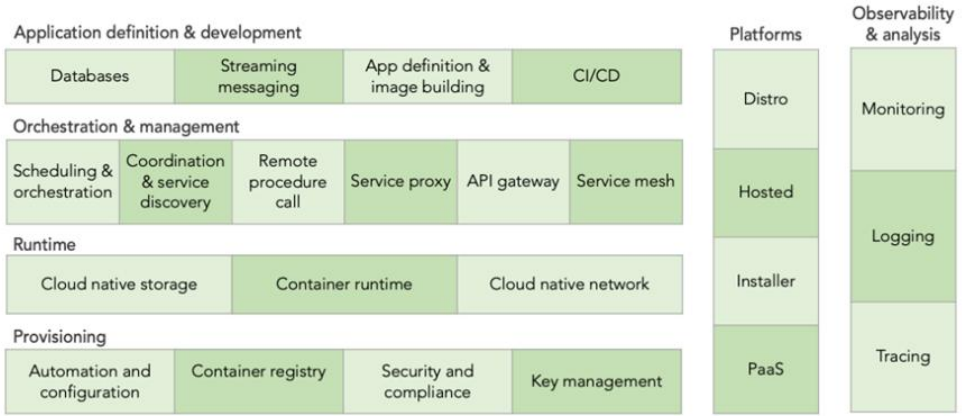
- Complexity in setup; requires working with a third-party connectivity provider.
- Requires high-bandwidth routers on-premises.

What is CNCF Landscape?

What are 6 Categories of CNCF?

Definition: The Cloud Native Computing Foundation (CNCF) Landscape is a visual representation of the various projects and tools that fall under the CNCF umbrella. It's designed to help navigate the ever-growing ecosystem of cloud-native technologies.

6 Categories of CNCF:



References

1. *7 elements of highly successful zero trust architecture | video - zscaler*. (2022, November 3).
2. Software AG. (n.d.). *SPIFFE*. Softwareag.com. Retrieved April 15, 2024, from https://techradar.softwareag.com/technology/spiffe-spire/
3. *Getting Started — envoy 1.30.0-dev-1095be documentation*. (n.d.). Envoyproxy.Io. Retrieved April 15, 2024, from https://www.envoyproxy.io/docs/envoy/latest/start/start
4. *Your first secret*. (n.d.). Your First Secret | Vault | HashiCorp Developer. Retrieved April 15, 2024, from https://developer.hashicorp.com/vault/tutorials/getting-started/getting-started-first-secret
5. *CNCF landscape*. (n.d.). Cncf.Io. Retrieved April 15, 2024, from https://landscape.cncf.io/
6. Cyber Law Toolkit. (n.d.). Microsoft Exchange Server data breach (2021). Retrieved from https://cyberlaw.ccdcoe.org/wiki/Microsoft_Exchange_Server_data_breach_(2021)