Technology Guide

Zero Trust - Zero Trust Reference Architecture

Authors

Xiang Wang

Heqing Zhu

Hongjun Ni

Yun Lan

Ju Huang

John DiGiglio

Michihiro Koyama

Pradeepsunder Ganesh

1 Introduction

Secure Access Service Edge (SASE) is becoming a major trend to deliver an architecture of converged security as a service on edge and cloud. As a critical function in SASE, Zero Trust Network Access (ZTNA) replaces legacy castle-and-moat approach (Perimeter Security), which builds a stack of network security functions in a predefined corporate perimeter. The enterprise network trend that remote workers, bring your own device (BYOD), and cloud-centric services are not located within an enterprise-owned network boundary is driving the deployment of ZTNA. It assumes least privilege and avoids lateral movement between resources. Enterprises are adopting zero trust solutions in a rapid pace. Gartner estimates that 60% of enterprises will phase out their remote access virtual private networks (VPN) in favor of ZTNA by 2023.

The National Institute of Standards and Technology (NIST) publishes a specification on zero trust architecture, which is the cornerstone of zero trust solutions currently. Intel has a broad set of security technologies that could optimize zero trust architecture with enhanced security and accelerated performance.

This paper introduces Zero Trust Reference Architecture (ZTRA) with Intel confidential computing technology, a reference design to realize the ZTNA standards, delivering core features including user authentication, service authorization, and secure network tunnel connection. ZTRA achieves security enforcement with Intel confidential computing technology (for example, Intel® Secure Guard Extensions (Intel® SGX)) and WireGuard tunnels as VPN service example. This reference also demonstrates the performance boost with crypto acceleration on Intel® Xeon® Scalable processors. This provides a solid reference on how to build a more secure and performant ZTNA system with the 3rd Gen Intel® Xeon® Scalable processor that fulfills the requirements of enterprise.

This document is part of the Network Transformation Experience Kits.

Table of Contents

Figures

Tables

Document Revision History

Revision	Date	Description
001	November 2022	Initial release.

1.1 Terminology

Table 1. Terminology

Abbreviation	Description
ACL	Access Control List
AES	Advanced Encryption Standard
AES-NI	Intel® Advanced Encryption Standard New Instructions
BYOD	Bring Your Own Device
Enclave	Ring 3 application software running inside the Intel SGX protections
IDP	Identity Providers
IPsec	Internet Protocol Security
ISA	Instruction Set Architecture
KMS	Key Management System
NIST	National Institute of Standards and Technology
OIDC	OpenID Connect Authentication Protocol
RBAC	Role Based Access Control
SASE	Secure Access Service Edge
VPN	Virtual Private Networks
VPP	Vector Packet Processing
ZTNA	Zero Trust Network Access

1.2 Reference Documentation

Table 2. Reference Documents

Reference	Source
Intel® Xeon® Scalable Platform Built for Most Sensitive Workloads	https://www.intc.com/news-events/press-releases/detail/1423/intel-xeon-scalable-platform-built-for-most-sensitive
Crypto Acceleration: Enabling a Path to the Future of Computing	https://newsroom.intel.com/articles/crypto-acceleration-enabling-path-future-computing
Golang	https://go.dev/
HashiCorp Vault	https://www.Vaultproject.io/ https://medium.com/hashicorp-engineering/hashicorp-Vault-performance-benchmark-13d0ea7b703f
Occlum	https://github.com/occlum/occlum
Intel SGX Programming Reference and SDK for Linux	https://software.intel.com/content/www/us/en/develop/articles/intel-sdm.html#combined https://download.01.org/intel-sgx/latest/linux-latest/docs/ https://github.com/intel/linux-sgx
National Institute of Standards and Technology FIPS Publication 197, Advanced Encryption Standard (AES)	https://csrc.nist.gov/publications/detail/fips/197/final
3rd Generation Intel® Xeon® Scalable Processor - Achieving 1 Tbps IPsec with Intel® Advanced Vector Extensions 512 (Intel® AVX-512) Technology Guide	https://networkbuilders.intel.com/solutionslibrary/3rd-generation-intel-xeon-scalable-processor-achieving-1-tbps-ipsec-with-intel-advanced-vector-extensions-512-technology-guide
Create Intel SGX VM in the Azure portal	https://docs.microsoft.com/en-us/azure/confidential-computing/quick-create-portal
Intel® Software Guard Extensions (Intel® SGX) – Key Management Reference Application (KMRA) on Intel® Xeon® Processors Technology Guide	https://networkbuilders.intel.com/solutionslibrary/intel-sgx-kmra-on-intel-xeon-processors-technology-guide
Intel® Software Guard Extensions (Intel® SGX) - Key Management Reference Application (KMRA) on Intel® Xeon® Scalable Processors User Guide	https://networkbuilders.intel.com/solutionslibrary/intel-sgx-kmra-on-intel-xeon-processors-user-guide
Intel® Software Guard Extensions (Intel® SGX) – Securing Private Keys in an Encrypted Enclave for Your Service Mesh Demo	https://networkbuilders.intel.com/intel-software-guard-extensions-intel-sgx-securing-private-keys-in-an-encrypted-enclave-for-your-service-mesh-demo
Intel® AVX-512 and Intel® QAT - Accelerate WireGuard Processing with Intel® Xeon® D-2700 Processor Technology Guide	https://networkbuilders.intel.com/solutionslibrary/intel-avx-512-and-intel-qat-accelerate-wireguard-processing-with-intel-xeon-d-2700-processor-technology-guide

2 Overview

A zero trust system shown in Figure 1 typically consists of endpoint client, controller (policy decision point), secure gateway (policy enforcement point), and remote service. Client needs to build a connection with controller for identity authentication and service authorization. Controller integrates with third party identity providers (IDP) for user authentication. It then generates policies based on user's identity and attributes to allow least privilege access to remote services. As a policy enforcement point, the secure gateway in edge POP (Point of Presence) or cloud will pull access policies from controller and establish a secure tunnel with client. Each network access from client will be evaluated by the secure gateway before reaching to the remote service.

There are challenges in building such a reliable zero trust system:

1) Secrets are hard to protect. The core requirements of zero trust are authentication and authorization, which involve generation and management of many credentials, including encryption keys, tokens, passwords, certificates, etc. It's critical to protect secrets at rest, in transit and in use.
2) Encrypted network tunnels are expensive. Every communication channel must be encrypted, including clients to edge and even edge to cloud services. In encrypted network tunnels, asymmetric and symmetric crypto are compute intensive.
3) Access controls are complex. Access evaluation must happen on every network access to achieve strict service micro-segmentation. It requires lots of policy definitions and heavy process of role-based access control.

Intel security technologies bring added values to tackle these challenges.

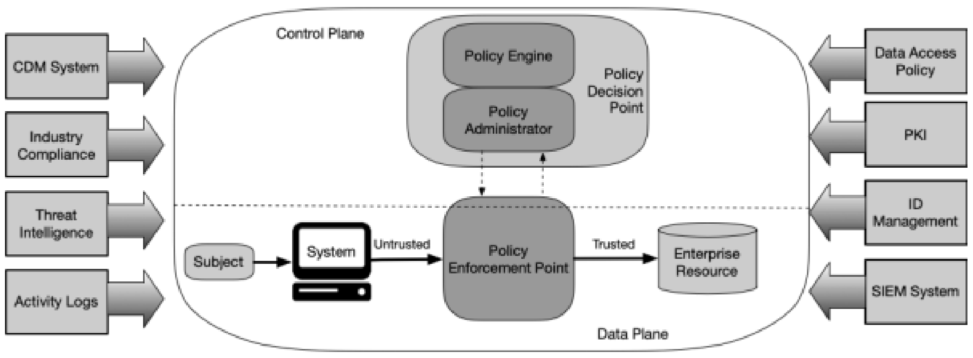

Figure 1. Zero Trust Architecture by NIST

3 Intel® Processor Technology

3.1 3rd Gen Intel® Xeon® Scalable Processor and Intel® Xeon® D-2700/1700 Processors

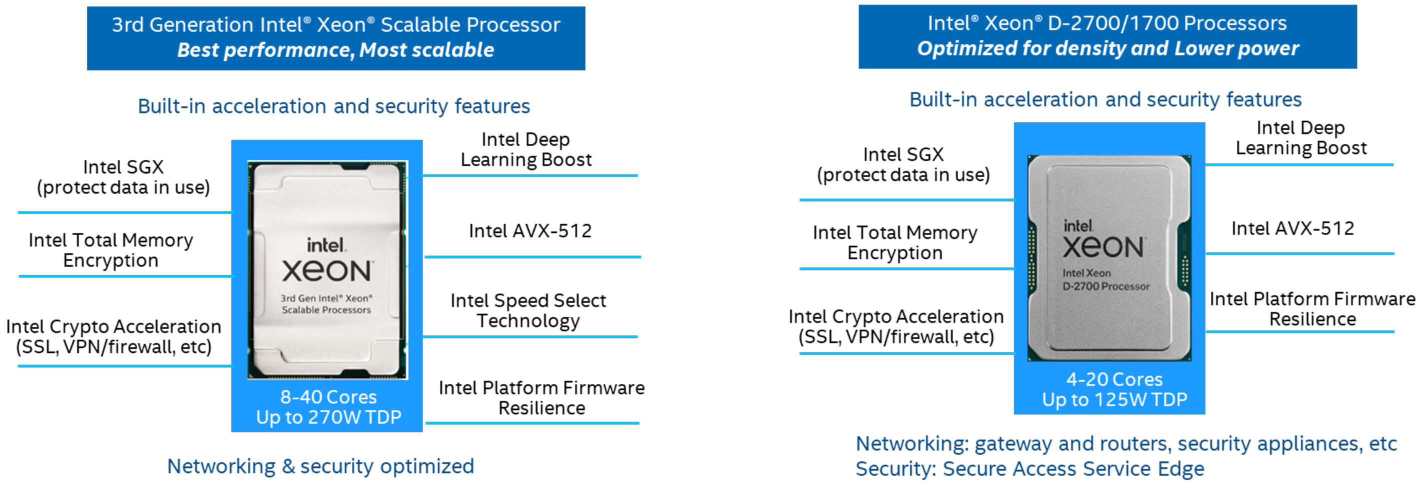
3rd Generation Intel® Xeon® Scalable Processor
Best performance, Most scalable
Built-in acceleration and security features
Intel SGX (protect data in use)
Intel Total Memory Encryption
Intel Crypto Acceleration (SSL, VPN/firewall, etc)
Intel Deep Learning Boost
Intel AVX-512
Intel Speed Select Technology
Intel Platform Firmware Resilience
intel. XEON 3rd Gen Intel® Xeon® Scalable Processors
8-40 Cores
Up to 270W TDP
Networking & security optimized
Intel® Xeon® D-2700/1700 Processors
Optimized for density and Lower power
Built-in acceleration and security features
Intel SGX (protect data in use)
Intel Total Memory Encryption
Intel Crypto Acceleration (SSL, VPN/firewall, etc)
Intel Deep Learning Boost
Intel AVX-512
Intel Platform Firmware Resilience
intel XEON Intel Xeon D-2700 Processor
4-20 Cores
Up to 125W TDP
Networking: gateway and routers, security appliances, etc
Security: Secure Access Service Edge

Figure 2. Intel® Xeon® Processors

As shown in Figure 2, 3rd Gen Intel Xeon Scalable processor delivers scalable performance for networking and security workloads. It adds security features useful for zero trust compared with prior generation. Intel SGX secures data in memory, enabling confidential computing. A key challenge in zero trust is to secure credentials (encryption keys, certificate, tokens, etc.). Intel SGX brings security enforcement to protect these secrets. New vector Intel® Advanced Encryption Standard New Instructions (AES-NI) and Intel® Advanced Vector Extensions 512 (Intel® AVX-512) instructions significantly boost crypto performance in use cases such as SSL, VPN, and firewall. Secure tunnels are compute intensive in zero trust. These new instructions could accelerate performance of zero trust system. Intel® Deep Learning Boost technology with Vector Neural Network Instructions (VNNI) improves inference performance for deep learning workloads. As zero trust solutions start to embrace AI for intelligent user to application segmentation, these build-in AI acceleration features could benefit zero trust solutions. Overall, deploying zero trust solution with 3rd Gen Intel Xeon Scalable processor allows security and performance in place.

Intel® Xeon® D-2700/1700 processors are SOC (System on Chip) designed to meet network edge usage models. They deliver performance and security with better power and space efficiency. Advanced security features are available in Intel Xeon D-2700/1700 Processors, including Intel SGX, crypto acceleration, Intel AVX-512, AI acceleration, etc. Their typical use cases include network and security appliances, SASE, etc. Since ZTNA is a critical function in SASE, Intel Xeon D-2700/1700 processors are also platform of choice for zero trust deployment.

3.2 Intel Confidential Computing Technology

Since zero trust system has strict data privacy and stringent security requirements, protecting data in transit and at rest is common practice, secure data in use (in memory) is not possible until Intel introduced the 3rd Gen Intel Xeon Scalable processor (formerly code named Ice Lake). 3rd Gen Intel Xeon Scalable processor introduces Intel SGX to standard off the shelf servers. This innovation allows the software to secure data in use.

Intel SGX offers hardware-based memory encryption that isolates specific application code and data in trusted execution environment (TEE). Intel SGX allows user-level code to allocate private regions of memory, called enclaves, which are designed to be protected from processes running at higher privilege levels. Intel SGX offers protection with a smaller available attack surface within the system. As shown in Figure 3, the enclave memory cannot be read or written from outside the enclave regardless of current privilege level and CPU mode, for example, operating systems, virtual machine managers, etc.

Intel SGX needs to go through attestation process to verify the hosting platform. Remote attestation allows a remote party to check that the intended software is securely running within an enclave on a system with the Intel SGX enabled. Intel® Software Guard Extensions Data Center Attestation Primitives (Intel® SGX DCAP) delivers the attestation service support for the enterprise and cloud service providers.

The library operating system (LibOS) technology enables users to run native application software in an Intel SGX enclave without extra modification efforts. Popular open-source LibOS solutions include Gramine and Occlum. By keeping credentials (private keys, tokens, personal identity information, etc.) in an enclave, zero trust system avoids the risk of exposing secrets to compromised users and external attackers.

At Intel, security comes first. For security technologies like Intel SGX, we integrate security principles at every lifecycle stage to help ensure that products are built with security in mind. We deliver discovery through offensive security research, Product Security Incident Response (PSIRT) and bug bounty. There is routinely sharing of security mitigations and updates.

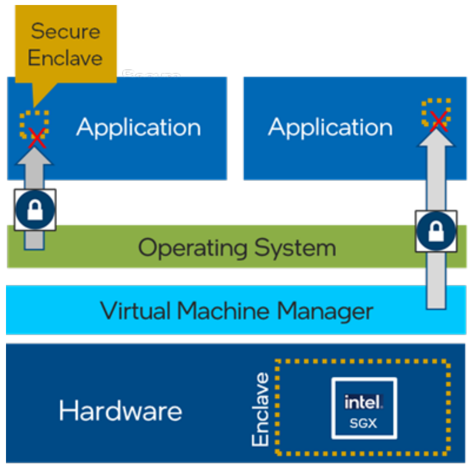

Figure 3. Intel® SGX Security Model

3.3 Intel Crypto Technology

Secure tunnel is a critical requirement in a zero trust system. The communication between client and secure gateway needs to rely on tunnels, such as SSL, IPsec, and WireGuard. The advanced solutions even establish another tunnel between secure gateway and remote services and then stitch two separate tunnels together to enforce stronger security. Crypto operations are at the core of secure data in transit and at rest. Crypto operations are compute intensive. In this reference design, the secure tunnels are shown as service example. This example also includes a high performant WireGuard solution. Latest instructions of Intel® Advanced Vector Extensions 512 (Intel® AVX-512) on 3rd Gen Intel Xeon Scalable processors accelerate Chacha20-Poly1305 algorithms used by WireGuard protocol significantly. The Intel AVX-512 parallel computation on multiple 64-byte data blocks for encryption and multiple 16-byte data blocks for authentication drives WireGuard implementation multiple times faster (https://networkbuilders.intel.com/solutionslibrary/intel-avx-512-and-intel-qat-accelerate-wireguard-processing-with-intel-xeon-d-2700-processor-technology-guide).

4 Open-Source Software Technology

4.1 HashiCorp Vault

HashiCorp Vault is a widely used open-source software, known as a secrets management system written in Golang. In modern system, secrets including API encryption keys, passwords, and certificates are used everywhere. Without automated access control and life cycle management for these secrets, it exposes high risks of credential lost and system malfunction. The design of Vault is to address these issues:

- Secrets management: Vault stores, generates, or encrypts data with various secrets engines. It requires seal and unseal mechanism for root key protection. To further strength security, there is an advanced secret life cycle management function to lease, renew, and revoke secrets automatically. Vault creates a lease with a time duration, renewability, etc. Once the lease is expired, Vault can automatically revoke the data.
- Authentication: User is authenticated against an auth method. A token is generated and associated to a policy.
- Vault can validate the client against third-party trusted sources, such as Microsoft Azure AD via OpenID Connect (OIDC) protocol. In this case, Client needs to own a valid user account in Azure AD. During authentication, Vault will forward a URL where the client should input credentials to prove his identity. Azure AD will notify Vault with an access token on successful authentication. Our zero trust reference architecture leverages Vault and Azure AD integration to implement user authentication.
- Authorization: Policies provide a declarative way to grant or forbid access to certain paths and operations in Vault. Vault grants access to secrets, keys, and encryption capabilities by issuing a token based on policies associated with client's identity. The client can then use their Vault token for future operations.
- Audit: Vault keeps a detailed log of all requests and responses. The audit log contains every authenticated interaction with Vault, including errors.

In a zero trust system, Vault can be used to provide the below critical security functions:

- Network tunnel key management: zero trust system relies on secure tunnel to exchange network traffic between client and secure gateway in edge pop or cloud. This involves intensive usage of public key encryption and thus private keys for tunnel establishment. Vault can act as a key management system to protect private keys from exposure.
- User identity authentication: authentication is a core requirement in zero trust. Vault has different authentication methods and native integration with third party IDPs. In all cases, Vault will enforce authentication as part of the request processing and then assign as set of policies based on user identity. Vault will delegate the authentication administration and decision to the relevant configured external auth method (for example, Amazon Web Services, GitHub, Google Cloud Platform, Microsoft Azure, etc.).
- Disk encryption: mature zero trust system needs to secure data at rest. Vault can protect encryption key for disk encryption operations.

Vault delivers secret encryption, but it does not guarantee the security of data in use. It still exposes secrets and encryption keys in memory when conducting compute operations. This creates a potential loophole that attackers could use. Protecting the Vault by running it in an Intel SGX closes this loophole.

4.2 Headscale and Tailscale

Tailscale is an open-source software designed mainly for VPN services. It enables encrypted point-to-point connections using WireGuard protocol. The overlay network it establishes makes every device within it communicates efficiently and securely. It has a wide support for different operating systems and native integration of security features, including identity validation with IDP, multi-factor authentication (MFA), policy enforcement with role-based access control (RBAC), logging and monitoring, etc.

Headscale is an open-source control plane solution designed to pair with Tailscale. It establishes communication channels with Tailscale nodes in the private overlay network. For all Tailscale nodes, Headscale fulfills the tasks including node authentication and registration, IP address assignment, peer sharing, access control policy creation based on node attributes (user group, namespace, device tag), etc.

The combination of Headscale and Tailscale delivers a set of baseline features (such as authentication and authorization) related to Zero Trust definitions. By leveraging them as the foundation, we can develop more fine-grained security features and faster network traffic processing speed.

4.3 Vector Packet Processing

Vector Packet Processing (VPP) is an open-source software framework, which is a major subproject of the Linux Foundation backed Fast Data Project (FD.io). VPP is a data plane software library for L2-L4 network packet processing. It divides network processing pipeline into multiple function stages represented as graph nodes. The design principle of VPP is batch packet processing, which accumulates packets for a dedicated graph node and then bursts them all at once. This improves CPU instruction cache efficiency and increases packet processing performance significantly.

VPP is a framework that builds the foundation for network data plane processing. Furthermore, Intel has dedicated efforts on optimizing VPP performance with Intel technologies. The combination of Intel Multi-Buffer Crypto for IPSec Library and Intel QAT delivers performance boost for protocols including IPsec and WireGuard. Intel AVX-512 instruction set increases access control list (ACL) algorithm performance with higher data parallelism. VPP is a good choice as the data plane engine in zero trust solution. Zero trust fundamental features including secure tunnel and access control can easily consume existing highly optimized functions in VPP on Intel processors.

5 Zero Trust Reference Architecture

ZTRA achieves zero trust capability by delivering core features including user authentication, service authorization, and secure tunnel. Traditional VPN only establishes a secure tunnel, which only requires one-time user identity check during establishment. All subsequent connections from the endpoint are trusted by default. This creates large attack surface that compromised endpoints could still access resources and make lateral movements. ZTRA conducts user identity authentication with authorized third-party identity providers and conducts continuous authorization by evaluating every connection initiated by endpoints with well-defined policies.

The system consists of endpoint client, controller, secure gateway (in edge server) and remote service as shown in Figure 4. The lightweight agent in the client establishes a connection with controller for authentication and authorization. It also sends requests to secure gateway via a secure WireGuard tunnel. Controller integrates with third party identity providers (IDP) via Vault for user authentication. It implements role based access control (RABC) by generating policies based on users' access privilege to each specific service and distributing policy to secure gateway for enforcement. Secure gateway receives requests from client, applies access control and forwards the traffic to remote services if granted. All connections to the controller are secured by HTTPs. It uses VPP for all data plane functions.

3rd Gen Intel Xeon Scalable processor and Intel Xeon D-2700/1700 processors are platforms to run ZTRA as they deliver advanced security features and high performance for zero trust systems. Based on them, we apply Intel optimizations to ZTRA. All credentials (private key, etc.) in both controller and secure gateway are stored in the enclaves protected by Intel SGX. WireGuard tunnels are accelerated by Intel AVX-512. ZTRA is delivered as a software package under Intel Proprietary License (IPL).

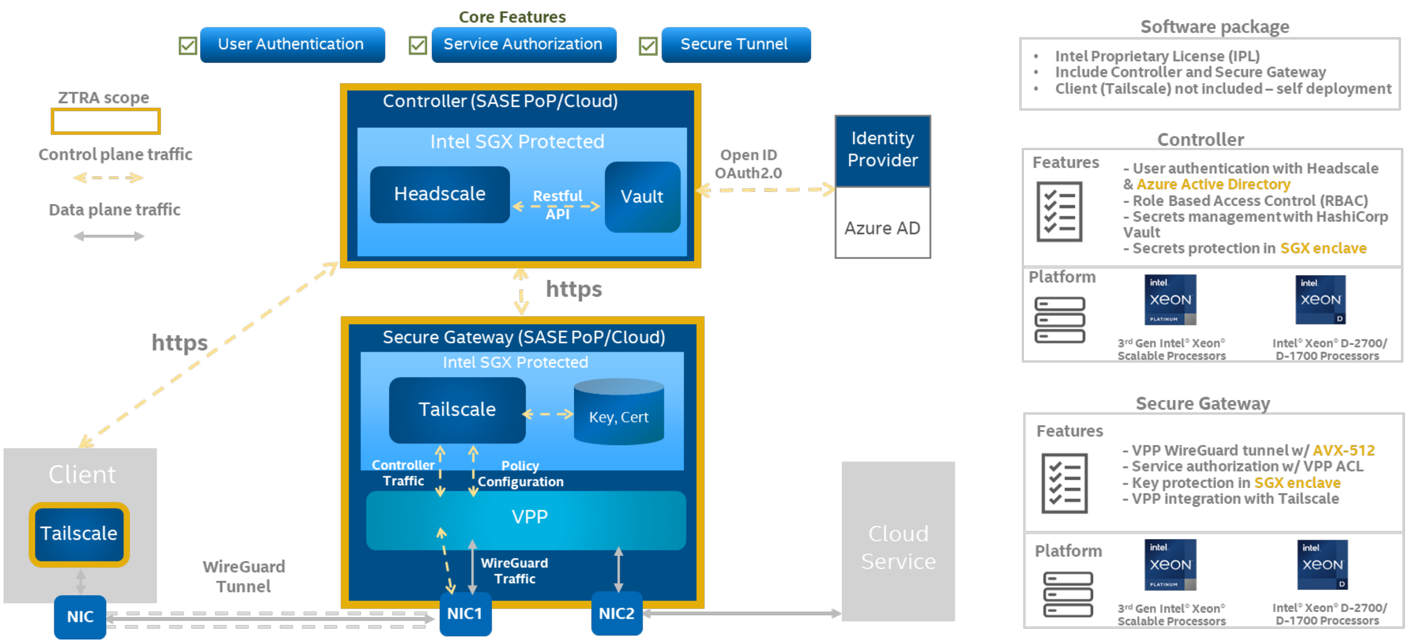
Core Features
User Authentication
Service Authorization
Secure Tunnel
ZTRA scope
Control plane traffic
Data plane traffic
Controller (SASE PoP/Cloud)
Intel SGX Protected
Headscale
Restful API
Vault
Open ID OAuth2.0
Identity Provider
Azure AD
https
Secure Gateway (SASE PoP/Cloud)
Intel SGX Protected
Tailscale
Key, Cert
Controller Traffic
Policy Configuration
VPP
WireGuard Traffic
NIC1
NIC2
Client
Tailscale
NIC
WireGuard Tunnel
Cloud Service
Software package
Intel Proprietary License (IPL)
Include Controller and Secure Gateway
Client (Tailscale) not included – self deployment
Controller
Features
- User authentication with Headscale & Azure Active Directory
- Role Based Access Control (RBAC)
- Secrets management with HashiCorp Vault
- Secrets protection in SGX enclave
Platform
3rd Gen Intel® Xeon® Scalable Processors
Intel® Xeon® D-2700/ D-1700 Processors
Secure Gateway
Features
- VPP WireGuard tunnel w/ AVX-512
- Service authorization w/ VPP ACL
- Key protection in SGX enclave
- VPP integration with Tailscale
Platform
3rd Gen Intel® Xeon® Scalable Processors
Intel® Xeon® D-2700/ D-1700 Processors

Figure 4. ZTRA System Architecture Overview

5.1 User Authentication

ZTRA uses Vault as the interface for authentication as it has native integration with IDPs including Azure AD via OIDC protocol. To pass user authentication, each user needs to have a valid account in the IDP before authentication. The authentication consists of two stages – controller application (APP) registration and user authentication.

Figure 5 shows Controller APP Registration workflow:

1. Controller administrator registers an APP account with an IDP.
2. Controller configures APPID, APPSecret, tokenURL, authURL, redirectURL, etc. in Vault.

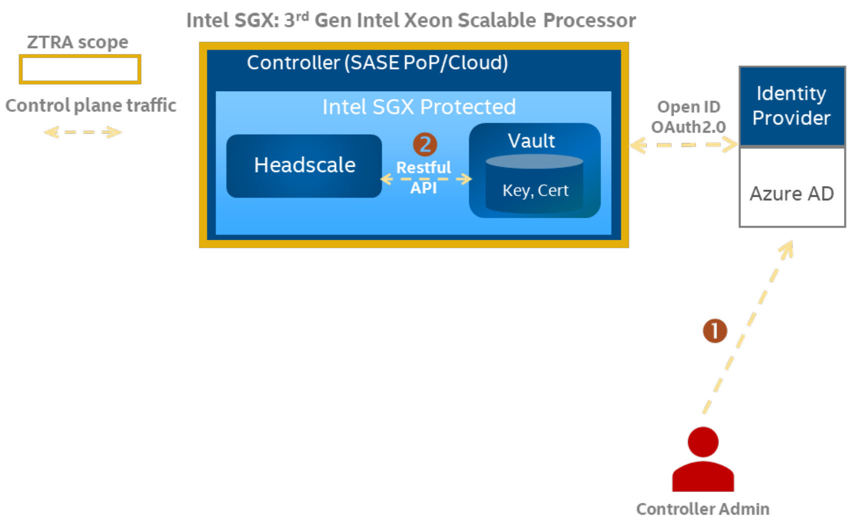

Figure 5. Controller APP Registration Workflow

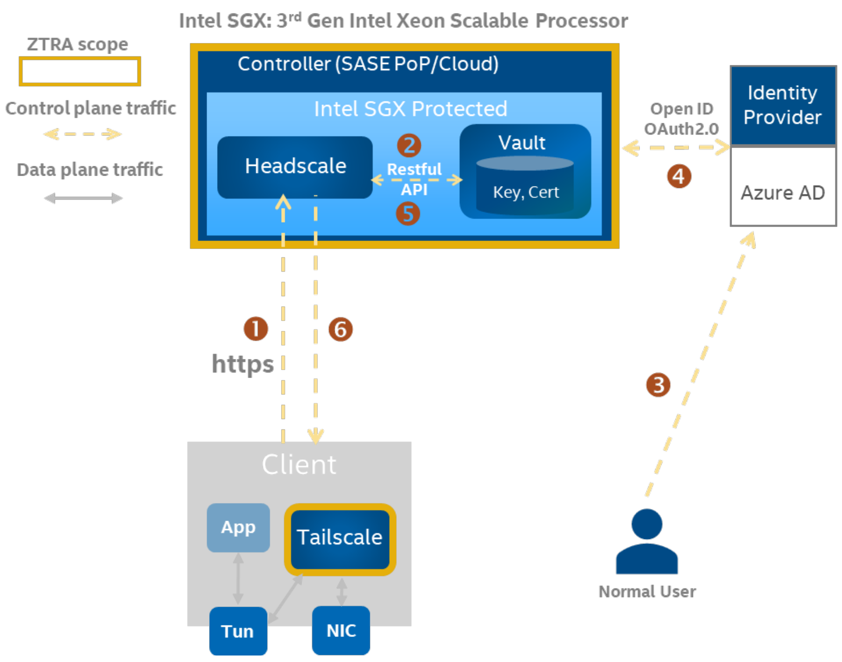

Figure 6. User Authentication Workflow

Figure 6 shows user authentication workflow:

1. Client sends authentication request with machine public key.
2. Vault generates an authentication URL (https://authurl?redirect_uri=http://redirect_ip:redirect_port/odic/callback&scope=xxx& response_type=code &state=xxx&appid=xxx.).
3. User visits the authentication URL, provides identity information and approves the authentication.
4. Vault receives the OIDC callback and visits authentication endpoint (https://tokeurl?code=xxx&appid=xxx&appsecret=xxx& grant_type=authorization_code) to get an access token.
5. Controller receives access token from Vault and saves client node as registered.
6. Client receives successful authentication response from Controller.

5.2 Service Authorization

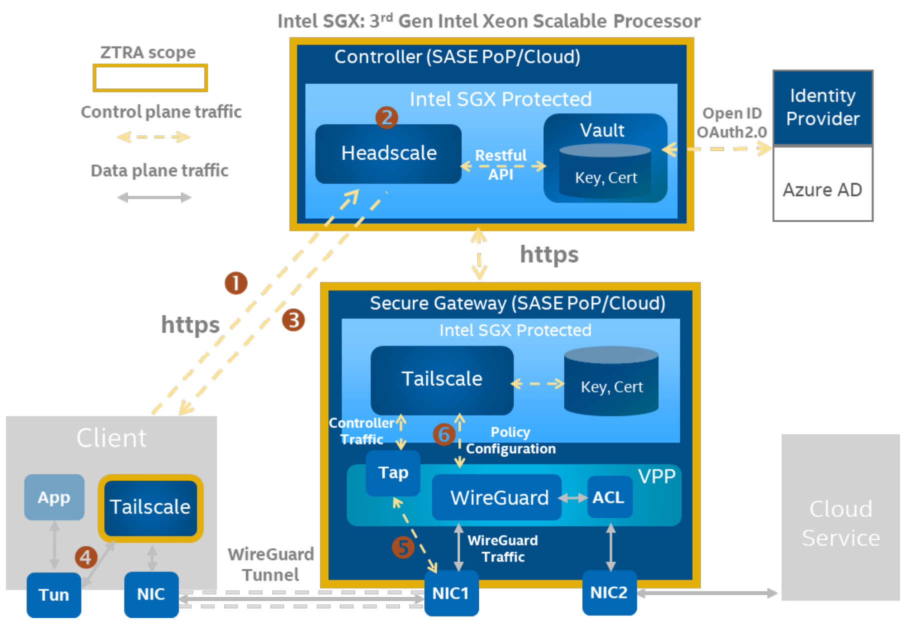

Figure 7. Service Authorization Workflow

ZTRA service authorization ensures least privilege access to services. It defines service access policy based on collected user identity and device tags upon successful authentication. Secure gateway then pulls policies from controller and injects them to VPP ACL engine for policy enforcement on every request sent from client. This is different from traditional VPN that only checks user credential/password at connection setup stage and no further authorization is ever requested. Optimizations of Intel AVX-512 on VPP ACL significantly boost matching performance.

Figure 7 shows service authorization workflow:

1. Client sends a service request to Controller.
2. Controller verifies Client identity (with public key) to grant if the user has the right to access this service. If granted, Controller then generates ACL rules.
3. Controller returns assigned IP, peer info, etc. in the response to a valid Client.
4. Client configures TUN interface with IP, routes, etc.
5. Controller passes configuration messages (peer info, ACL, etc.) to agent via a TAP interface in VPP.
6. Agent configures ACL rules in VPP.

```
{
 "acls": [
  {
   "Action": "accept",
   "Users": ["tag:webdevice", "group:sre"],
   "Ports": ["tag:webserver"]
  }
  // Other access rules here...
 ]
 // Other policy configuration here...
}
```

Figure 8. RBAC Policy Example

The policy definition is based on Role Based Access Control (RBAC):

1. ZTRA leverages a combination of keywords to define the policy:
    a. Action indicates whether to accept or deny the request.
    b. Users and Ports describe source and destination of the request.
    c. Group defines a cluster of users, hostname, namespace, etc.
    d. Tag represents attributes assigned to client device.

    These definitions meet core requirements of identity and attribute-based access control in zero trust system. Figure 8 shows a policy example where only a device with "webdevice" attribute or a user from the group of "sre" can access web server services.
2. Policy is an abstract representation for access control. Controller needs to decode each policy into ACL rules in {srcIP, dstIP, port, proto} format. It then distributes these ACL rules to secure gateway for policy enforcement.
3. Access control policy in zero trust system can be dynamic. New policies could be generated based on identity and attributes analysis of clients. To satisfy the requirement of real time policy update, secure gateway keeps a long live connection with Controller. Controller will push ACL rule updates to secure gateway automatically.

5.3 Data Plane Processing

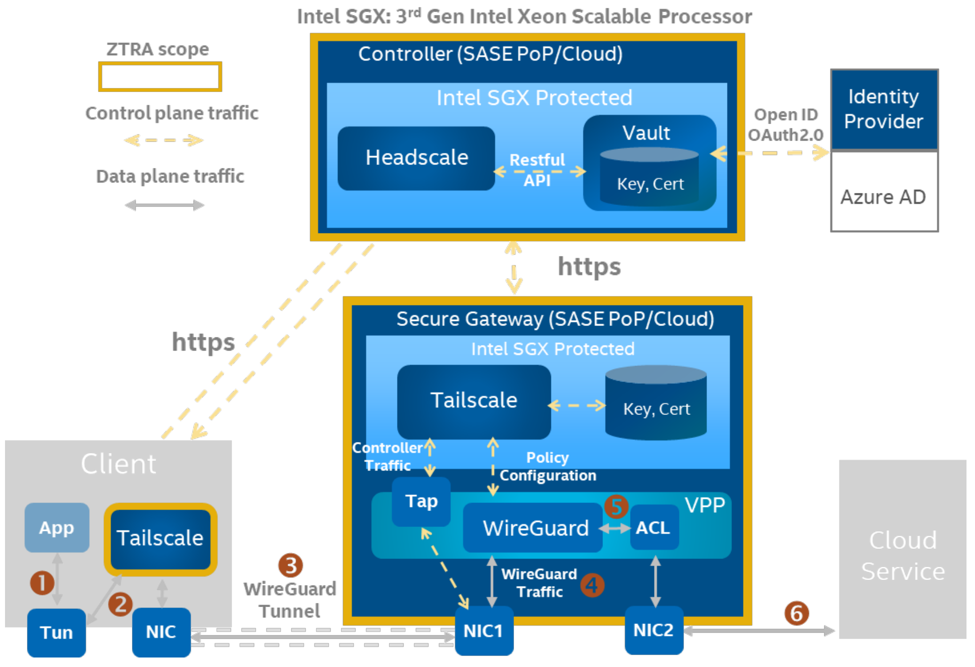

Figure 9. Service Request Workflow

Each service request from client needs to go through secure gateway for access control. Once it is granted, secure gateway will forward the request to remote service. This is different from a traditional VPN where once the session is secured, each access is not any longer under security check. Even if an access is compromised, the hacker cannot laterally move through the network because his subsequent access will still have to go through access control of ZTRA. Figure 9 shows the workflow for a service request.

In client device:

1. Endpoint app sends a service request via a configured TUN device.
2. TUN device forwards the traffic to Agent.
3. Agent sends the request to secure gateway by establishing a WireGuard tunnel.

In secure gateway:

4. VPP WireGuard (accelerated by Intel AVX-512) receives the traffic from the endpoint.
5. VPP enforces ACL policy for service authorization.
6. VPP forwards the traffic to remote service only when the client has appropriate access privilege

6 ZTRA Deployment

This section shows the basic deployment of ZTRA solution. It covers detailed install instructions for core components in ZTRA.

6.1 Deployment Setup

6.1.1 Intel® SGX with Occlum Setup

1. Intel SGX setup for Vault

Occlum Golang toolchain only supports go1.16.3. We chose Vault v1.8.9 for compatibility reason. As Vault needs to make external https connections, we build Occlum instance by using host certificate in /etc/ssl/certs/. We also should set appropriate memory size.

Below shows how to build Vault Occlum instance.

Step 1. Compile the user program with the Occlum toolchain.

```
# cd vault_v1.8.9
# occlum-go build -o bin/vault
```

Step 2. Generate a secure Occlum FS image.

```
# vi copy.yaml:
```

```
includes:
  - base.yaml
targets:
  - target: /bin
    copy:
      - files:
        - ../bin/vault
  - target: /
    mkdirs:
      - etc/ssl
  - target: /etc/ssl
    copy:
      - from: /etc/ssl/
        dirs:
          - certs

# occlum new vault_instance
# cd vault_instance
# new_json="$(jq '.resource_limits.user_space_size = "2560MB" |
            .resource_limits.kernel_space_heap_size="320MB" |
            .resource_limits.kernel_space_stack_size="10MB" |
            .resource_limits.max_num_of_threads=128 |
            .process.default_stack_size = "40MB" |
            .process.default_heap_size = "320MB" |
            .process.default_mmap_size = "960MB" ' Occlum.json)"
# echo "${new_json}" > Occlum.json
# rm -rf image
# copy_bom -f ../_copy.yaml --root image --include-dir /opt/occlum/etc/template
```

Step 3. Run the user program inside an Intel SGX enclave.

```
 occlum run /bin/vault server -dev
```

2. Intel SGX setup for controller.

All setup steps are similar to the prior section for Vault. We must set additional VAULT_TOKEN and VAULT_ADDR environment variables to Occlum enclave for controller:

```
# new_json="$(jq '.resource_limits.user_space_size = "2560MB" |
      .resource_limits.kernel_space_heap_size="320MB" |
      .resource_limits.kernel_space_stack_size="10MB" |
      .resource_limits.max_num_of_threads=128 |
      .process.default_stack_size = "40MB" |
      .process.default_heap_size = "320MB" |
      .process.default_mmap_size = "960MB" |
      .env.untrusted[0] = "VAULT_TOKEN" |
      .env.untrusted[1] = "VAULT_ADDR" ' Occlum.json)"
# echo "${new_json}" > Occlum.json
```

6.1.2 Vault Setup

After Vault startup, we need to enable our secrets path and set OIDC config and role using API or CLI. Below is an example of CLI configuration.

```
# vault secrets enable -path=ztra kv
# vault auth enable oidc
# vault write auth/oidc/config <<EOF
{
"oidc_client_id":"xxx",
"oidc_client_secret":"xxx",
"default_role":"dev",
"oidc_discovery_url":https://login.microsoftonline.com/d500132f-f214-4452-9c06-
b9b55e35e40f/v2.0
}
EOF

# vault write auth/oidc/role/dev <<EOF
{
  "user_claim": "email",
  "oidc_scopes": ["profile", "email"],
  "allowed_redirect_uris":"${VAULT_ADDR}/ui/vault/auth/oidc/oidc/callback",
  "policies":"ztrakvreader",
  "claim_mappings":    {"email": "email", "ipaddr": "loginip"}
}
EOF
```

6.1.3 VPP Setup

For the VPP setup in Figure 9, secure gateway uses two interfaces with one connects public network and the other connects to server subnet. VPP's linux-cp plugin is used for passing local traffic between Linux kernel and VPP through TAP interface.

```
# vi /etc/vpp/startup.conf
dpdk {

        dev 0000:4b:01.3 {
                name vppeth0
        }

        dev 0000:4b:11.0 {
                name vppeth1
        }
}

# lcp create vppeth0 host-if eth0
# set interface ip addr vppeth0 100.100.5.5/24
# set interface state vppeth0 up
# lcp lcp-sync on

# set interface ip addr vppeth1 192.168.1.2/16
# set interface state vppeth1 up
```

7 ZTRA Software Availability

ZTRA builds a good reference implementation of zero trust system. It differentiates with other solutions by applying unique Intel security technologies including confidential computing and crypto acceleration. The first release (22.06) covers all features including user authentication with Azure AD, service authorization with RABC, secure WireGuard tunnel and secrets protection with Intel SGX. To access this software, please contact your Intel sales partner, or the document authors.

8 Summary

Embracing zero trust is a necessary trend driven by the digital transformation of enterprise IT. Selecting the appropriate hardware platform (such as 3rd Gen Intel Xeon Scalable processor) and choosing the appropriate software design can reduce attack surface with strong security posture, increase agility without restrictions imposed by rigid perimeter, and improve control over services migrated to cloud environment. Building a robust zero trust system is full of challenges. This document demonstrates ZTRA as a foundational zero trust network access framework and provides best practices to strengthen it with security offerings from Intel Xeon Scalable processors. The combination of Intel confidential computing feature (Intel SGX) and Vault provides comprehensive protection for credentials including private key, token, etc. Crypto acceleration with Intel AVX-512 instructions and Intel QAT further accelerates secure tunnel connections in the zero trust system. These features give insights on building and upgrading your zero trust systems by maximizing values of Intel technologies.

Appendix A Platform Configuration

Name	Inspur NF5280M6
Vendor	Inspur
Product Name	Server Machine
BIOS Version	06.00.01
SGX	Yes
SGX EPC size	2GB (max 64GB)
OS	Ubuntu 20.04.3 LTS
Kernel	5.16.0-rc4
IRQ Balance	enabled
CPU Model	Intel® Xeon® Platinum 8358 @ 2.60GHz
Base Frequency	2.6GHz
CPU Family	6
CPU Model	106
CPU(s)	128
Thread(s) per Core	2
Core(s) per Socket	32
Socket(s)	2
NUMA Node(s)	2
Turbo	enable
Memory Installed	512GB

Software Configuration	Software version	Location
Host OS	Ubuntu 20.04.3 LTS	https://ubuntu.com/
Kernel	5.16.0-RC4	https://www.kernel.org/
Vault	Vault 1.8.9	https://www.vaultproject.io/
Occlum	Occlum 0.26.3	https://occlum.io/
VPP	stable/2202	https://fd.io/
Headscale	0.15.0	https://github.com/juanfont/headscale/
Tailscale	1.22.2	https://github.com/tailscale/tailscale

1122/DN/WIPRO/PDF 751287-001US