



# Seven Elements of Highly Successful Zero Trust Architecture

An Architect's Guide to the  
Zscaler Zero Trust Exchange

## Authors

Nathan Howe, VP, Emerging Technology & 5G, Zscaler

Sanjit Ganguli, VP, Transformation Strategy & Field CTO, Zscaler

Gerard Festa, VP, Product Marketing, Zscaler



## About the Authors

Nathan Howe, (VP, Emerging Technology & 5G), Sanjit Ganguli, (VP, Transformation Strategy & Field CTO), and Gerard Festa, (VP of Product Marketing) bring substantial insights on Zero Trust with careers spanning the globe and companies including Gartner, Aruba, Nestle, Riverbed, Cisco, and Verizon. Their leadership and innovative view on cloud, security, transformation, and emerging technologies make them the authorities in modern zero trust direction.

Nathan Howe and Sanjit Ganguli have previously authored, [\*The 7 Pitfalls To Avoid When Selecting An SSE Solution\*](#).

# Contents

<b>An Overview of Zero Trust</b>	<b>2</b>
<b>Connecting to the Zero Trust Exchange</b>	<b>21</b>
<b>Section 1 – Verify</b>	<b>25</b>
Element 1: Who is connecting?	26
Element 2: What is the access context?	37
Element 3: Where is the connection going?	48
<b>Section 2 – Control</b>	<b>70</b>
Element 4: Assess Risk (adaptive control)	72
Element 5: Prevent Compromise	81
Element 6: Prevent Data Loss	99
<b>Section 3 – Enforce</b>	<b>112</b>
Element 7: Enforce Policy	114
<b>Connecting to the Applications</b>	<b>123</b>
<b>A Fast, Reliable, and Easy-to-Operate Zero Trust Architecture</b>	<b>135</b>
<b>Getting Started with Your Zero Trust Journey</b>	<b>144</b>
<b>Appendix 1 – Application Segmentation Primer</b>	<b>150</b>

# An Overview of Zero Trust

Who is this guide for?

The definition of “zero trust” has been widely used and abused since being coined over a decade ago. This guide seeks to provide a clear definition of zero trust in the context of the Zscaler Zero Trust Exchange and to help readers understand how it can be architected. Network and security professionals overseeing zero trust initiatives should read this guide to learn how to deliver effective control and visibility across their zero trust initiatives.

# IT and security are becoming enablers of digital transformation

Businesses undertake digital transformation journeys to increase efficiency, improve agility, and achieve a competitive advantage. And, in today's economy, this transformation is accelerating. For the business to succeed, IT must first undergo a transformation of its own—one that typically starts with applications moving from data centers to the cloud, which in turn necessitates a network transformation. To transform the network, security must transform with it.



Figure 1: Secure digital transformation requires a combination of application, network, and security transformation.

For the past three decades, organizations have been building and optimizing complex, wide-area, hub-and-spoke networks for connecting branches and factories to applications in the data center. The network was secured with a stack of security appliances and firewalls using an architecture known as castle-and-moat security. This was so named because the security stack created a network perimeter (or moat) around the data center (or castle). This architecture prevented access to anyone outside the network, but granted privileges to anyone within. This network and security architecture served us reasonably well when applications lived in the data center and users worked from the office.



## An Overview of Zero Trust

In today's world, the workforce is dispersed and applications no longer reside solely in the data center. A large number of applications have moved to public clouds as SaaS, IaaS, or PaaS offerings, resulting in a hybrid environment. It no longer makes sense to backhaul traffic to the data center to securely reach applications in the cloud. For fast and productive collaboration, application traffic must go direct. No one wants to fly from New York to London with a stopover in Chicago.

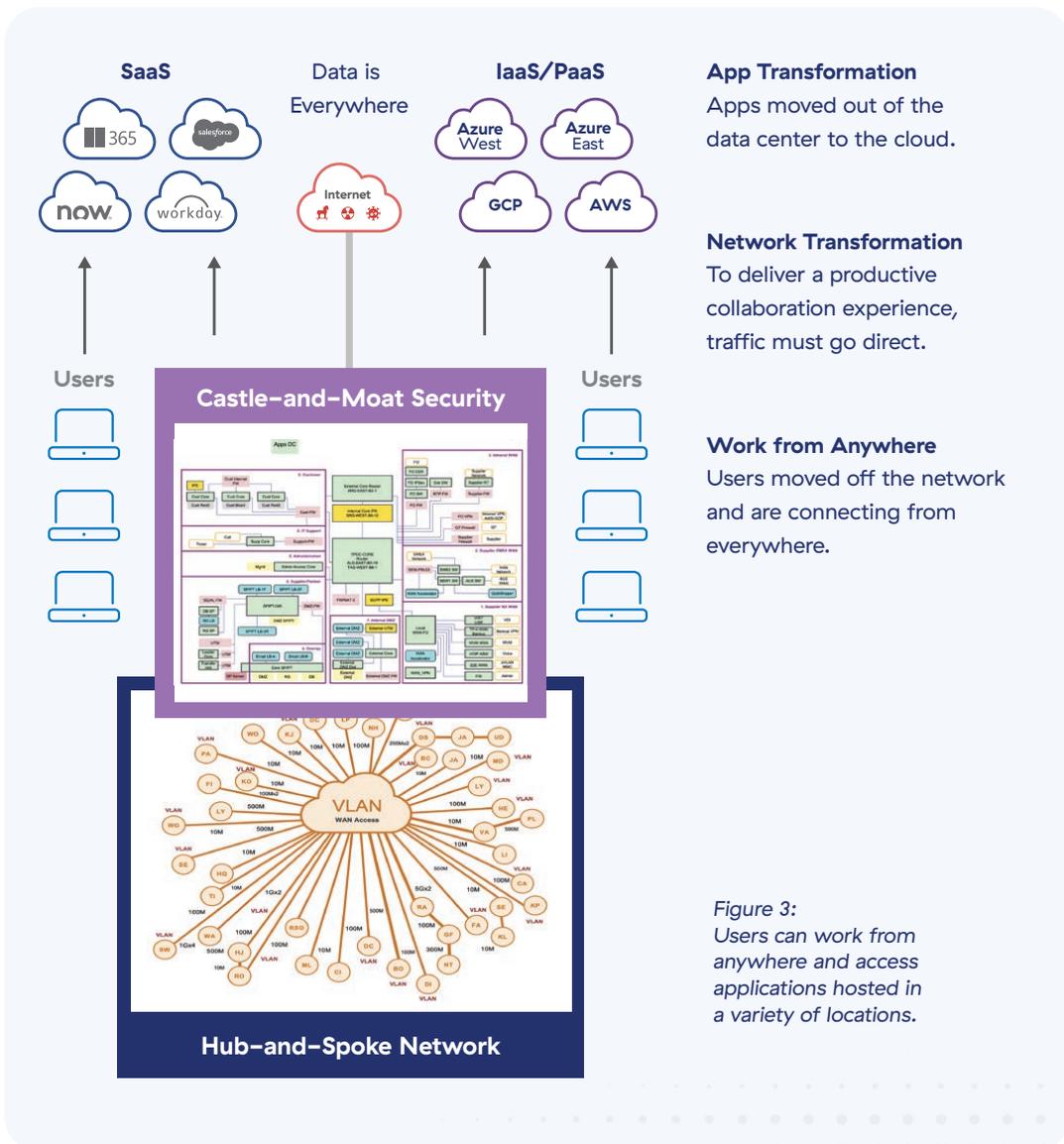


Figure 3: Users can work from anywhere and access applications hosted in a variety of locations.

Hub-and-spoke networks were built and optimized to link on-premises users with applications residing in data centers. Since the network and applications are intertwined, application access requires users, devices, and workloads to be connected to the corporate network. For a remote workforce, this means extending the network via VPN where each client is allocated a routable IP address on the enterprise network. Having 20,000 remote users means extending the network to 20,000 locations or homes via VPNs. These VPN termination points then become front doors anyone on the internet can discover and attack.

For internal applications hosted in IaaS and PaaS environments, the network must be further extended to all in-use cloud providers. Doing so allows attackers to inflict substantial impact on an enterprise in four steps.

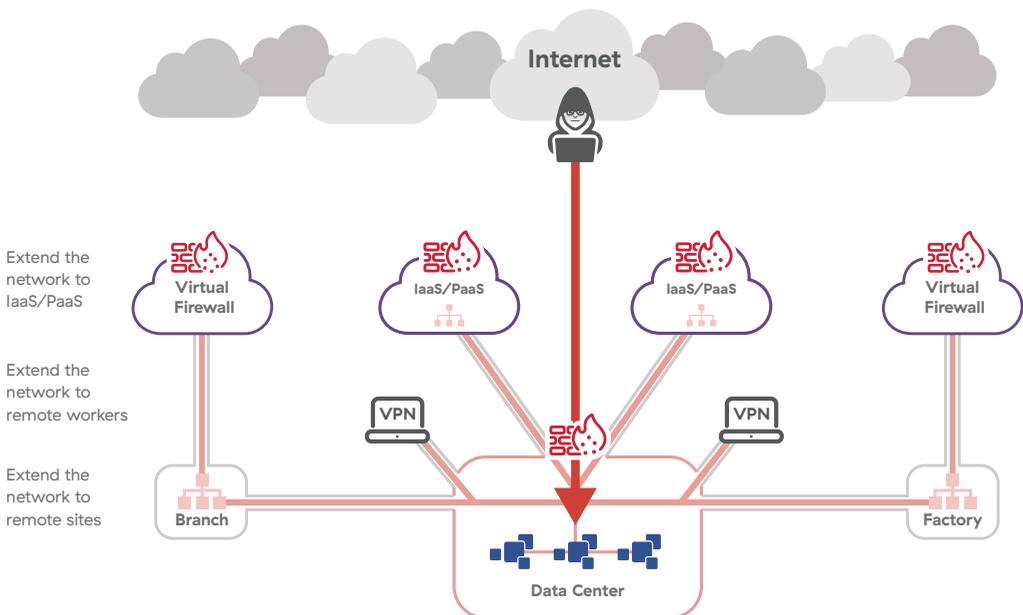


Figure 4: When application access requires network access, the network needs to be extended to where users, devices, and workloads are located.

## Four Steps to Breach an Enterprise:

### 1 They find your attack surface.

Every interconnected network has an implicit trust in that anyone who can access these networks can connect to any application residing on them. The shared network context, be it internet-based users connecting via VPN, workloads exposed for access (on any network), etc., ultimately leaves services open to receive a connection. The moment a service requires access from an initiator over a shared network, that service is exposed as an attack surface.

Hub-and-spoke networks have historically leveraged implicit trust to allow for connectivity, but the design also introduces performance problems when workforces and applications become distributed. To resolve this problem and its associated costs, many companies deployed local internet breakouts to route traffic directly. Virtual firewalls can be deployed to protect these breakouts, but this also increases their internet attack surface.

Every internet-facing service, including firewalls—whether in the data center, cloud, or branch—can be discovered, attacked, and exploited. Remember, firewalls connect networks and attempt to apply controls at that network layer. The moment access is needed, an ingress network listener must be exposed. Subsequently, this exposed network IP listener is open to anyone and anything that shares this network, which could ultimately be the entire internet.

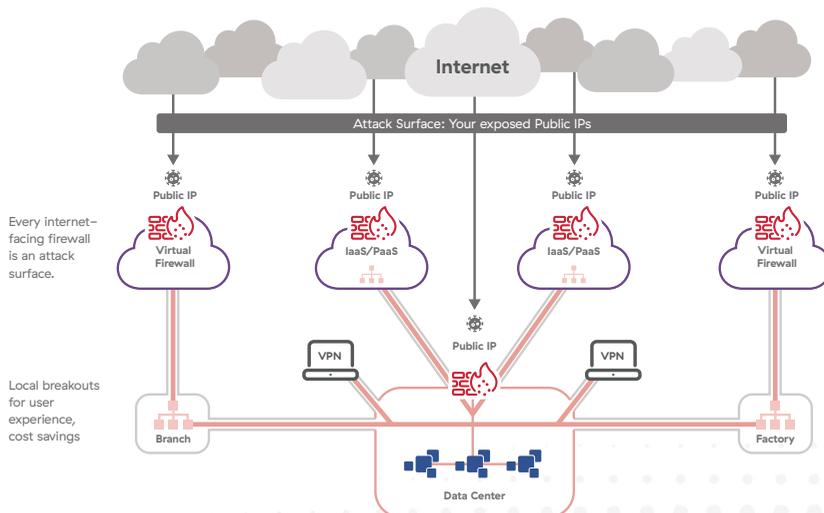


Figure 5: They find you. Everything exposed to the internet is your attack surface.

## 2 They compromise you.

Cybercriminals bypass conventional detection methods by exploiting the trust of common services. Attackers either directly target your exposed services (e.g., firewalls, VPNs, workloads) or entice end users by hosting malicious content. Firewalls and antivirus appliances, which once provided adequate protection, are anchored in a centralized network control point that hasn't kept up with the pace and sophistication of modern users, apps, and modern-day attacks. It is not a matter of if you will be compromised, but when. With an exposed attack surface, the organization is subject to both randomized and targeted attacks.

Attackers identify and target a corporation's weakest links to access its network. Once inside, they establish a beachhead, ensure they have multiple paths back into the network should the original entry point be secured, and begin scanning for high-value targets.

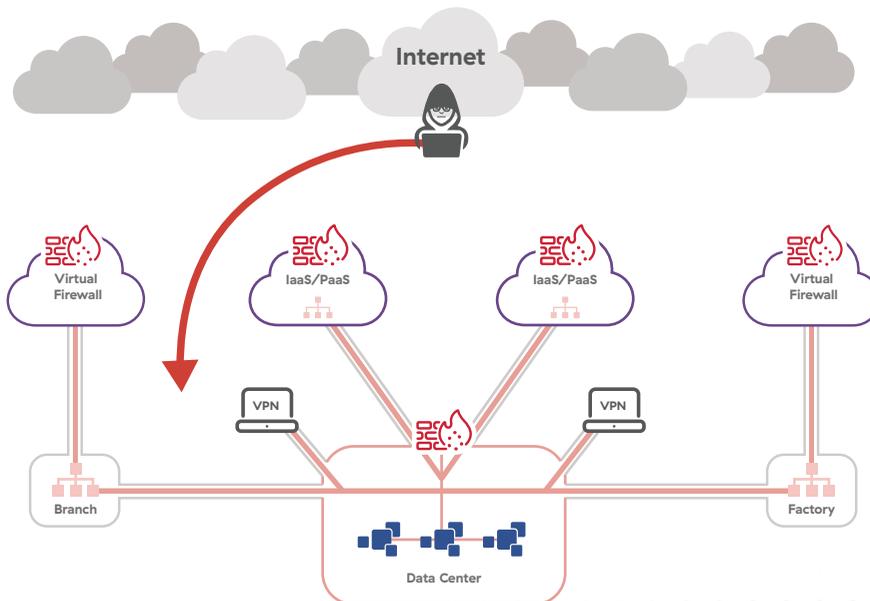


Figure 6: They compromise you, infecting users, devices, and workloads.

### 3 They move laterally.

Extending networks for added functionality based on the principle of a shared network context allows for easy access, as users and apps are both on the network. But it also provides the same easy access to infected machines since network-based controls have difficulty controlling lateral or east-west movement across the breached network. A single infected machine in a user's home—or an infected workload in a public cloud—that shares the trusted network context can access all applications, giving it the potential to cripple a business.

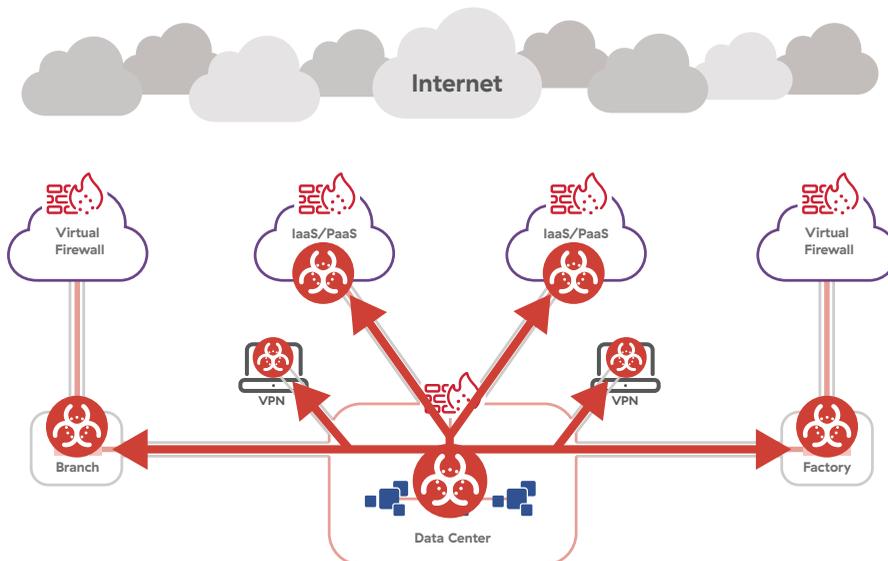


Figure 7: They move laterally, finding high-value targets for ransomware and other attacks.

## 4 They steal your data.

After discovering and exploiting high-value assets, attackers will attempt to leverage trusted services like SaaS, IaaS, and PaaS—as well as known and accepted protocols like standard HTTPS encryption—to set up backchannels and exfiltrate data. An example is the [Colonial Pipeline breach](#), where an attacker was able to use stolen VPN credentials to enter a corporate network, move laterally to access sensitive financial data and disrupt operations, and ultimately hold a piece of U.S. critical infrastructure for ransom—a practice known as ransomware.

### Ransomware at a glance

- **Extortion:** Attackers render enterprise information unusable and demand money for its return.
- **Double Extortion:** Attackers threaten to release enterprise information if not paid.
- **Triple Extortion:** Attackers leverage the stolen information to inflict additional damage, e.g., DoS of the customer or the sale of customer data in order to apply additional pressure.

Attackers continuously refine these tactics and have adopted double and sometimes triple extortion techniques to increase their chances of collecting payment by threatening to leak customer data or cripple operations.

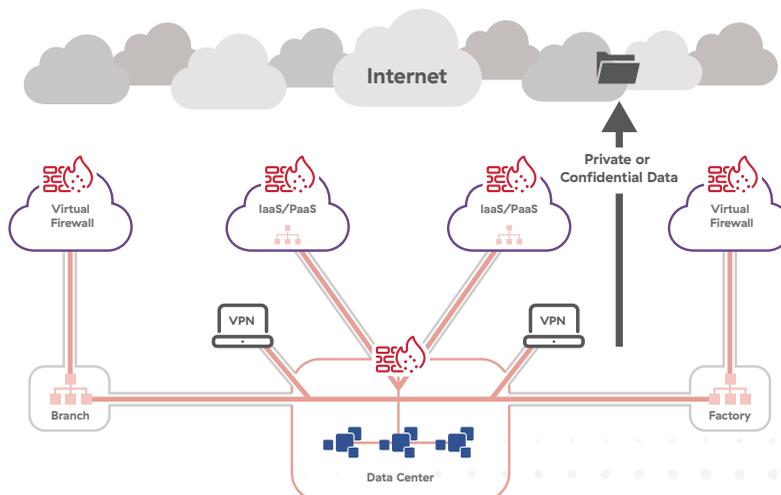


Figure 8: They steal your data and avoid firewall detection.

Vulnerabilities inherent to legacy network and security architectures highlight the imperative to evolve the design into something radically different that addresses modern day attacks and exposure. This evolution involves both a network and security transformation, enabling a more ubiquitous and granular policy construct, and at the same time enabling digital transformation. The answer is a zero trust architecture that removes the attack surface and provides secure connectivity between users/devices, IoT/OT devices, and workloads, wherever they may reside.

# Introducing Zero Trust Architecture

The challenges caused by legacy network and security architectures are pervasive and long-standing, and require a rethinking of how connectivity is granted in the modern world. This is where zero trust architecture must be leveraged—an architecture where no user or application is trusted by default. Zero trust is based on least-privileged access, which ensures that trust is only granted once identity and context are verified and policy checks are enforced.

[NIST](#) defines the underlying principle of a zero trust architecture as “no implicit trust granted to assets or user accounts based solely on their physical or network location (i.e., local area networks versus the internet) or based on asset ownership (enterprise or personally owned).” It’s an overhaul of the old proverb “Never trust. Always verify.”

This approach treats all network communications as hostile, where communications between users and workloads or among workloads themselves are blocked until validated by identity-based policies. This ensures that inappropriate access and lateral movement are prevented. This validation carries across any network environment, where the network location of an entity is no longer a factor and not reliant on rigid network segmentation.

Key architectural advantages of the zero trust approach versus legacy network security are summarized in the figure below:

	Legacy Network and Security Architecture	Zero Trust Architecture
Attack Surface	Firewalls/VPNs published on the internet Can be exploited, susceptible to DDoSed	Apps not exposed to the internet You can't attack what you can't see
Connection	App access requires corporate network access, allows lateral movement of users and threats	Connects a specific, authorized user to a specific, authorized resource
Proxy/Pass-through	<u>Firewall/Pass-through</u> Inspects a limited data buffer Unknown files pass through Alerts after infection	<u>Proxy</u> Full content inspection, including TLS/SSL Hold and inspect unknown files before reaching the endpoint
Tenancy	VMs of single-tenant appliances in a public cloud	Cloud-native, multitenant design like Salesforce/Workday

Figure 9: High-level comparison of legacy network architecture versus zero trust architecture.

To fully understand zero trust architecture, it's useful to break it down into individual building blocks (or elements) that are executed before any connection is established. These elements ensure that all enterprise services—user/devices, IoT/OT devices, and workloads—are subject to the same set of controls when requesting access to assets.

There are seven essential elements of zero trust architecture, grouped into three categories:



The full stack of control actions within a zero trust architecture can be found in Figure 10.

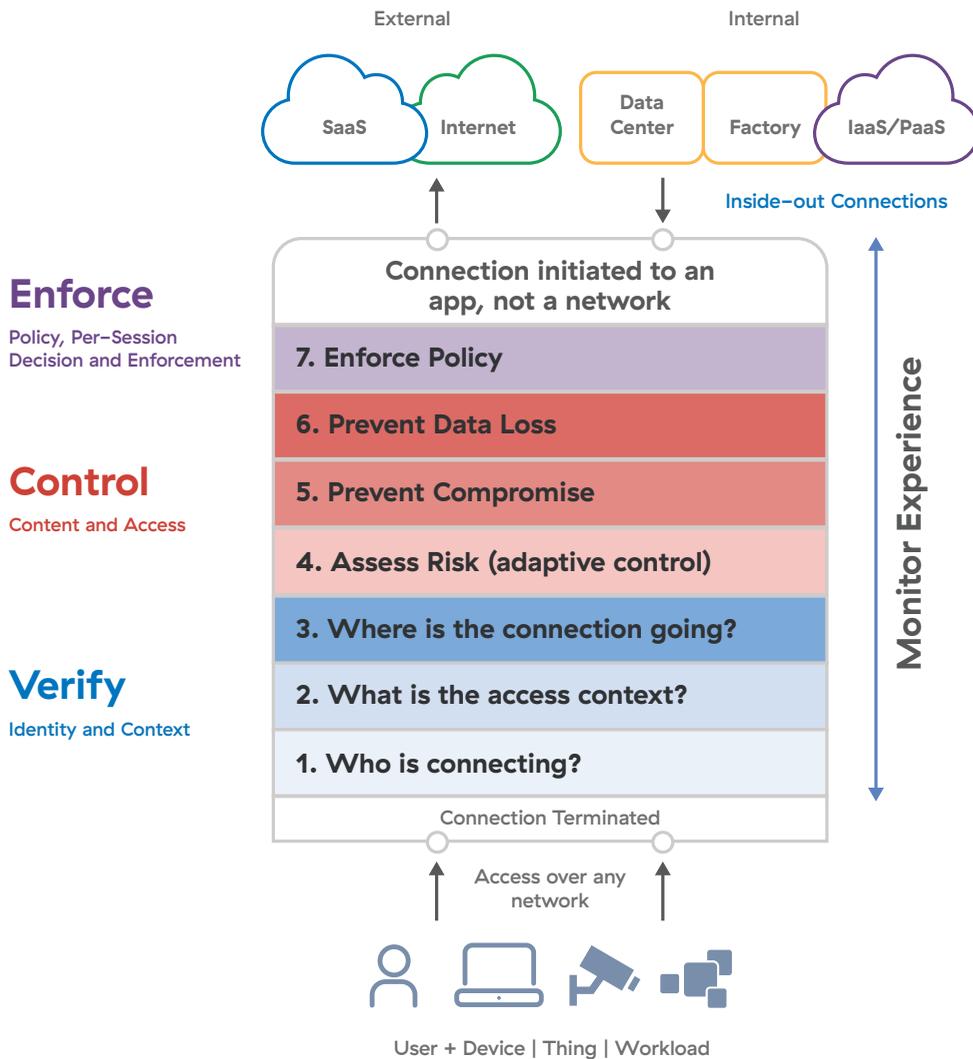


Figure 10: The seven elements of zero trust architecture.

## The following seven elements form the foundation of any zero trust design.

### Verify Identity and Context

Once the user/device, IoT/OT device, or workload requests a connection, irrespective of the underlying network, the zero trust architecture first terminates the connection and verifies identity and context by understanding the “who, what, and where” of the request:

- 1 Who is connecting?** — This first element verifies the identity of the user/device, IoT/OT device, or workload through integrations with third-party providers (IdPs) as part of an enterprise identity access management (IAM) provider.
- 2 What is the access context?** — This element validates the context of the connection requester, looking at attributes such as the role, responsibility, request time, location, and circumstances of the request. This profile data is collected from multiple sources, including IdP and third-party integrations.
- 3 Where is the connection going?** — This element confirms that the owner of the verified identity has the rights and meets the required context to access the requested application or resource based on segmentation rules. This entity-to-resource segmentation is the cornerstone of zero trust.

### Control Content and Access

Once identity and context of the requesting entity are verified and segmentation rules are applied, zero trust architecture then evaluates the risk associated with the connection request, as well as inspects the traffic for cyberthreats and sensitive data:

- 4 Assess risk** — This element leverages AI to dynamically compute a risk score for the user/device, IoT/OT device, or workload based on factors including device posture, threats, destination, behavior, and policy.
- 5 Prevent compromise** — This element utilizes inline decryption and deep content inspection of entity-to-resource traffic to identify and block malicious content.
- 6 Prevent data loss** — This element also uses decryption and deep content inspection of entity-to-resource traffic to identify sensitive data and prevent its exfiltration through inline controls or by isolating access within a controlled environment.

## Enforce Policy, Per-Session Decision and Enforcement

After verifying the identity/context and controlling for risk, policy is enforced before ultimately establishing a connection to the internal or external application:

- 7 Enforce policy** — This element uses the outputs of previous elements to determine what action to take regarding the requested connection. This action can take multiple forms, ultimately resulting in a conditional allow or conditional block.

Note that all seven elements may not be used for all policies or types of traffic. For certain applications, such as VoIP, the choice may be to not inspect the content.

Each element in this process feeds into the next, creating a dynamic decision tree that is utilized for each user/device, IoT/OT device, or workload-to-resource request. Every connection must evaluate identity, profile, user risk, site risk, posture, and content as criteria for deciding whether to grant access conditionally or to conditionally block (see Figure 11).

None of the elements above can come at the expense of the user experience. Zero trust architecture must therefore be able to monitor performance and diagnose experience issues to ensure that these seven elements do not put an undue burden on the user.

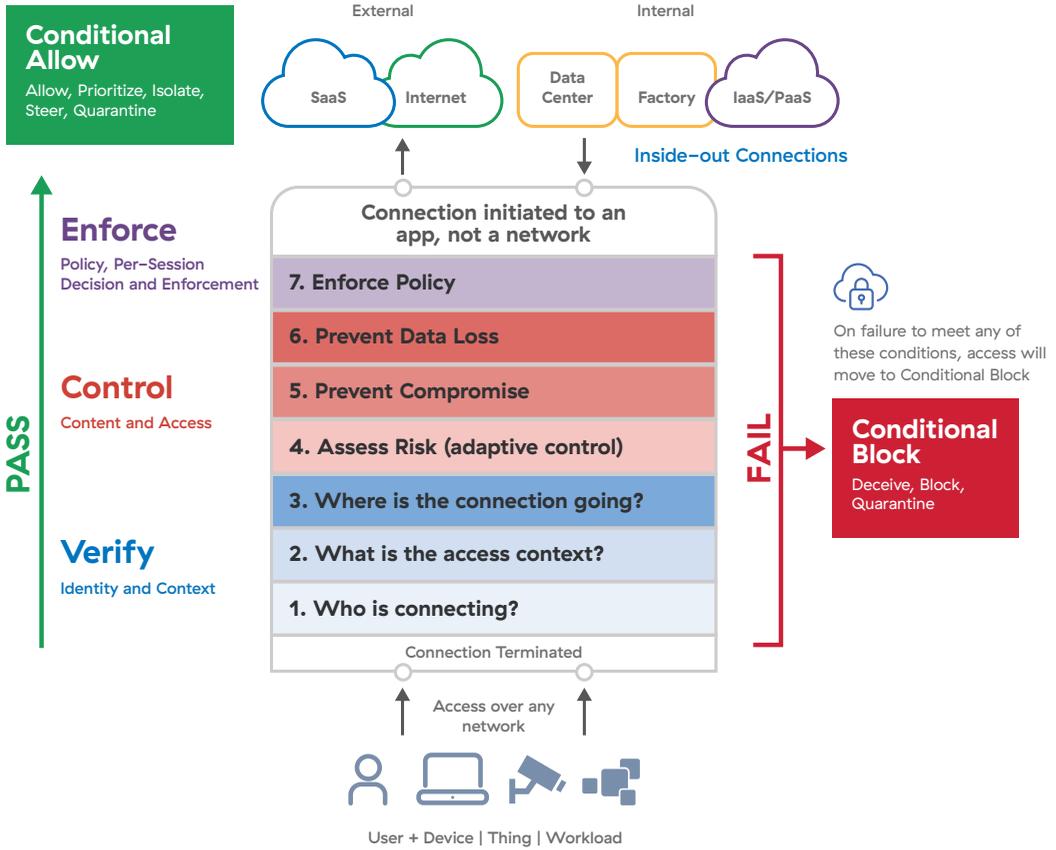


Figure 11: Allow or restrict decisions based on zero trust principles.

# Zscaler's Zero Trust Exchange

How is zero trust architecture implemented? Through Zscaler's Zero Trust Exchange. This integrated platform of services protects users and workloads using identity and context to securely broker user/device, IoT/OT, and workload communications over any network from any location. The Zero Trust Exchange architecture secures users, applications, and data, rather than the network.

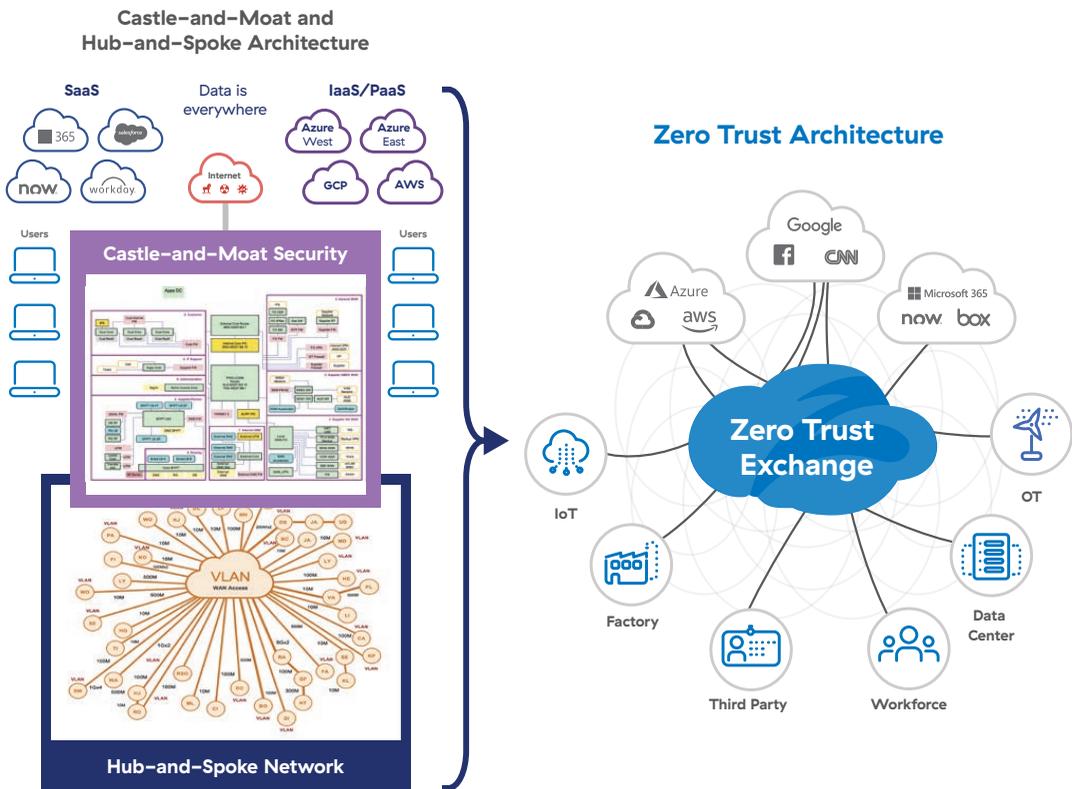


Figure 12: Transforming to the Zscaler Zero Trust Exchange.

## An Overview of Zero Trust

As seen in Figure 13, the Zscaler Zero Trust Exchange uses identity-based controls to enforce policies that securely provide user-to-workload, third-party access, workload-to-workload, and location-to-location segmentation. These zero trust connections are brokered by the Zero Trust Exchange without ever granting broad network access.

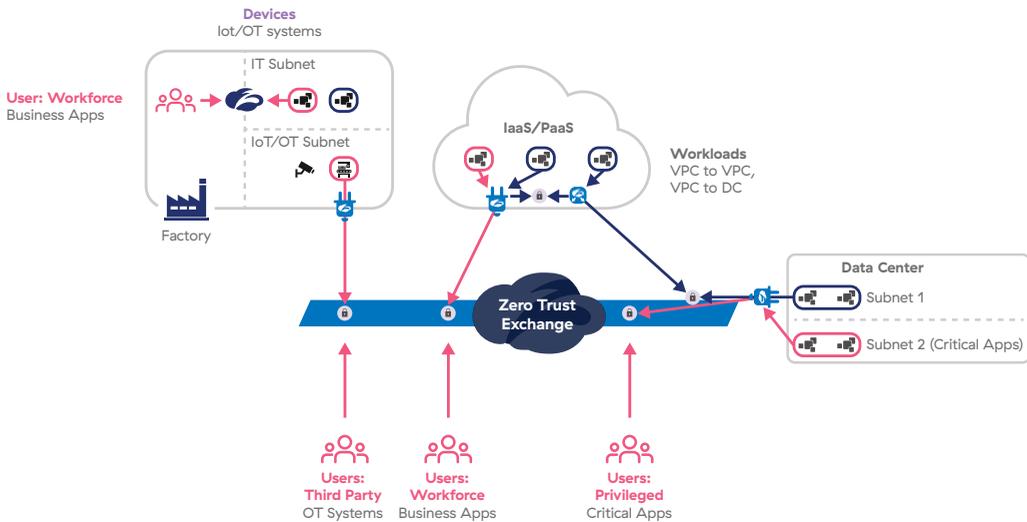


Figure 13: The Zscaler identity-driven segmentation model uses business policies to connect users/devices, IoT/OT devices, and workloads over any network to prevent lateral movement.

## Zero Trust, SSE, and SASE

The eBook [The 7 Pitfalls to Avoid When Selecting an SSE Solution](#) defines and describes Gartner's Security Service Edge (SSE) in detail, while touching on Zscaler's Zero Trust Exchange and its applicability to SSE and the broader Secure Access Service Edge (SASE).

So, how do the concepts of zero trust architecture, as described here, relate to the broader concepts of SSE? They are closely intertwined. Gartner's SSE provides a framework that combines the main elements of network security—including the Secure Web Gateway (SWG), Zero Trust Network Access (ZTNA), and a Cloud Access Security Broker (CASB), among other components—as provided from the cloud at a location near the end user. ZTNA in this context relates merely to user-to-private application access, according to Gartner's research.

Zero trust architecture, in this book, is a part of a much broader discussion beyond Gartner and NIST's narrower definitions. Zscaler has implemented zero trust as a core architectural component of the Zero Trust Exchange (the name gives it away), and it permeates through every element of the SSE framework. This includes a zero trust approach for users accessing any application (internal or external), IoT/OT connectivity, and workloads accessing resources in a multicloud environment or on the internet itself. It includes not only verification but also deep inspection and enforcement at every stage while dynamically controlling for risk.

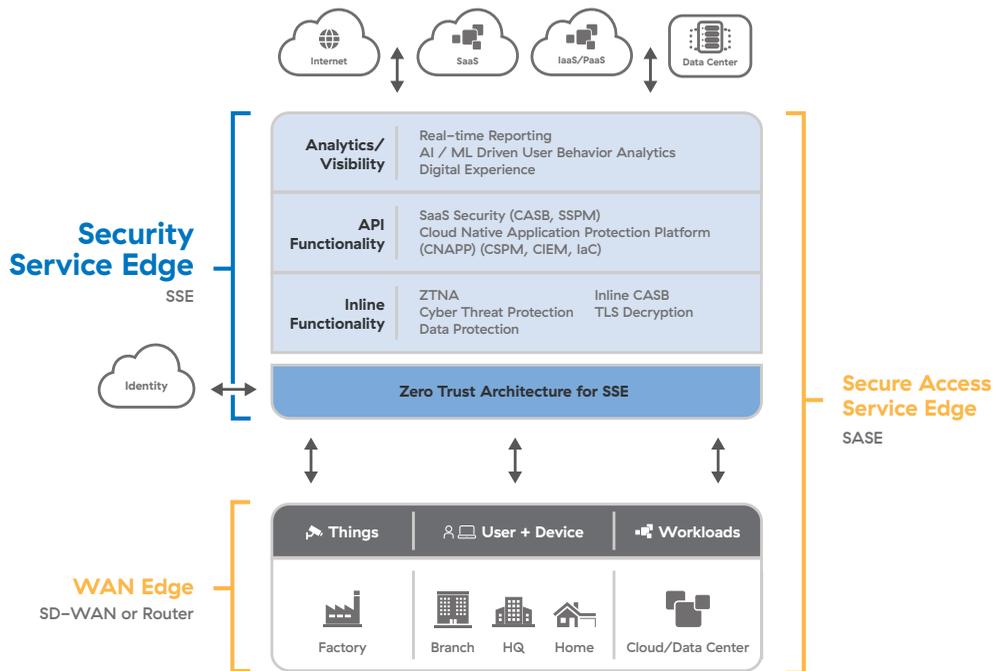


Figure 14: SSE/SASE and zero trust.

SSE is a subset of the larger SASE framework, also from Gartner, that includes WAN edge components (like SD-WAN) alongside SSE components. Zscaler’s network-agnostic platform works with any network underlay and leverages integrations with SD-WAN vendors to provide users with a seamless experience.

The remainder of this guide will explore each zero trust element in detail, discuss the technology required to accomplish it, highlight architectural considerations, and illustrate how each is accomplished within the Zscaler Zero Trust Exchange. For each element, we’ll follow two example users, John and Jane Doe, on their journey through the zero trust process of accessing applications, including progress reports tracking their progression.

# Connecting to the Zero Trust Exchange

# Connecting to the Zero Trust Exchange

Before discussing these seven critical elements, we must first explain how connections are established with the Zero Trust Exchange. Zero trust elements are enforced via a cloud-native and globally distributed set of POPs that comprise the Zero Trust Exchange. Users/devices, IoT/OT devices, and workloads must first establish a connection to this zero trust cloud platform, where it is subsequently terminated so security controls can be enforced.

The Zero Trust Exchange is a highly available and globally distributed service, so that connections are requested and enforced at the most effective location to ensure the best user experience. The Zero Trust Exchange can also be run wherever is most suitable for the enterprise, meaning that it can be within a customer’s premises, cloud, or edge platform. This brings the power of the Zero Trust Exchange as close to the consumer initiator as possible.

Zero trust connections are, by definition, independent of any network for control or trust. Zero trust ensures access is granted by never relying on a shared network between the originator (user/device, IoT/OT device, or workload) and the destination app. By keeping these separate, zero trust can be properly implemented and enforced over any network. The network can be located anywhere and be built on IPv4 or IPv6, since it is simply the means of connecting initiators to destination apps.

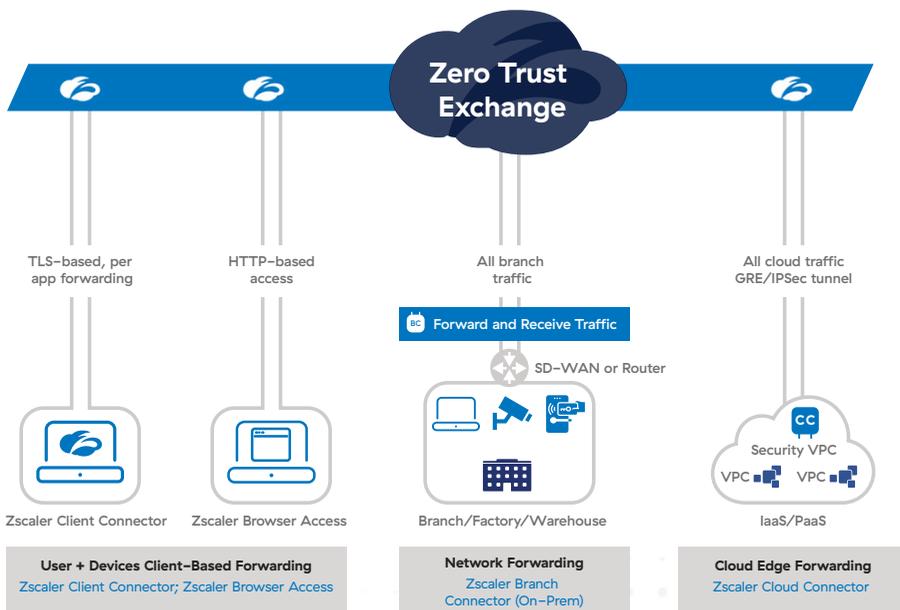


Figure 15: Representation of the mechanisms forwarding traffic to the Zscaler Zero Trust Exchange.

Zscaler's architecture allows secure connections to be made in a number of ways (see Figure 15). Most commonly, an agent called the Zscaler Client Connector is installed on the endpoint to create a tunnel-based connection to the Zero Trust Exchange for the protection of SaaS and internet-bound traffic. This same Client Connector also provides a persistent control plane and dynamic, microsegmented data plane tunnels to the Zero Trust Exchange for the purpose of internal app protection. Traffic is delivered to the application via a corresponding outbound-only data plane tunnel from the Zscaler App Connector: detailed explanation of this connectivity is outlined in the section Connecting to the Applications.

In scenarios where Client Connector cannot be deployed, customers can leverage site-based forwarding options (GRE or IPSEC tunnels) to connect entire sites to the Zero Trust Exchange. Note that connectivity to the Zero Trust Exchange works in conjunction with SD-WAN as well as traditional edge device tunnels. Since zero trust is a network-agnostic architecture, Zscaler maintains integrations with most SD-WAN vendors so that tunnels can be automatically provisioned from an SD-WAN site to the Zero Trust Exchange.

Site forwarding is meant to encompass all possible enterprise locations, from branches, offices, factories, and warehouses to IaaS cloud-hosted workloads, etc. Zscaler has additional form factors available to accommodate these enterprise locations.

The deployment of site-based forwarding is not mutually exclusive. Devices running Client Connector can exist and operate on the sites that forward traffic to the Zero Trust Exchange.

Zscaler Branch Connectors facilitate secure communication from these sites and can be deployed on-premises in locations like satellite offices and factories. Conversely, Cloud Connectors offer connection mechanisms from IaaS and PaaS locations, allowing protection for workload-to-workload (multicloud) and workload-to-internet connections. Both the Branch and Cloud Connector variations allow bidirectional, secure communication.

For unmanaged remote user devices, where an agent cannot be installed, DNS CNAME redirects traffic to a protected, private portal. Users then authenticate against an IdP to access web, RDP-based, and SSH-based applications. This is called Zscaler Browser Access, and does not require any explicit forwarding mechanism. This functionality prevents direct interaction with the services, while additional protection via browser isolation inherently prevents threats from reaching the user/server as well as provides data protection.

Once an inside-out connection is initiated with the Zero Trust Exchange, that connection is terminated as the Zero Trust Exchange acts as a forward proxy. This termination initiates the seven elements, as the connection is not pass-through. Once the elements are completed, a new connection is established between the Zero Trust Exchange and the application to complete the transaction.

## Zero Trust Progress Report

Throughout this architectural guide, a progress report will show the journey of two example users, Jane and John Doe. At each stage, their stats will be displayed as examples of their access requests, assessments, and the ultimate policy control applied through the Zero Trust Exchange.



*Progress Report: Starting at zero, Jane and John Doe request access to applications X and Y.*

# Section

# 1

# Verify

# Verify

Identity and Context

3. Where is the connection going?
2. What is the access context?
1. Who is connecting?

## Element 1: Who is connecting?

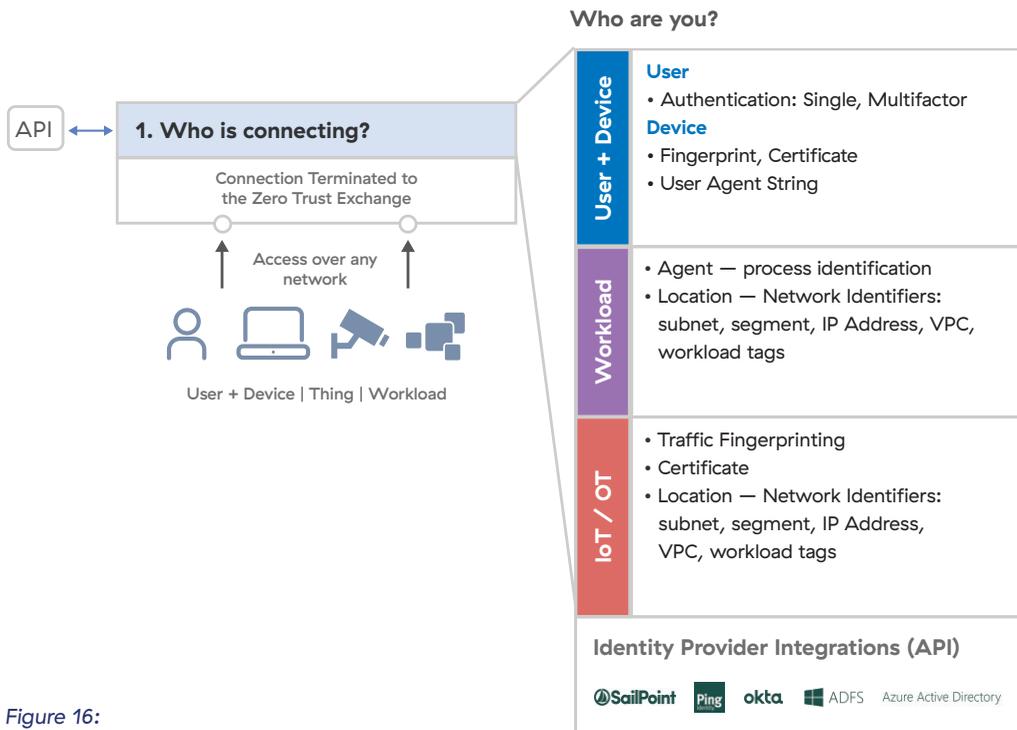


Figure 16:  
This element asks who you are, which is not a singular value. An identity needs to be made up of various values, not just a user's individual identity.

Common security wisdom states that “nothing, absolutely nothing, should ever implicitly happen.” No person or thing should be allowed to access or view anything, not even the front door of the building, without first being verified and their access rights assessed.

In other words, the requester must always be verified before access is granted. Each requester should be treated individually and only granted access to what their identity allows. It’s this initial identity verification that determines if a requester is able to proceed farther down the zero trust path.

Identity, as we know the term today, was coined by the French mathematician Marie Jean Antoine Nicolas de Caritat, Marquis de Condorcet (no chance of messing up their identity), when trying to understand the relationship between a person and the collective foundations of the systems the person was part of: officially “the quality of being identical.”

This idea of identity was formulated to understand the connection we have to groups that bind us. Today, at least outside of the enterprise, identity is a mechanism of individualization.

Within the enterprise, this definition of identity is especially apt. Employees are identified not only by who they are, but also assigned to groups that organize them. This alignment of identity and identity-based access led the computing industry to the revolutionary idea of least privilege—individuals should be granted access to specific resources based on their role’s specific needs.



### Why is this important?

The specific, granular nature of identity is the cornerstone of zero trust and the Zero Trust Exchange—not only for people, but also for devices, internet-enabled things, and workloads. These entities must present a valid identity to differentiate themselves in order to gain access to allowed resources via the correct set of controls. All other access should be blocked.

Zero trust architecture is designed so that access is completely blocked until explicitly allowed. This differs from most traditional environments where network access was implicitly granted and managed via antiquated network controls. When an initiator presents the correct identity, then access may be granted only to the specific set of allowed services and nothing more. This “never trust, always verify” approach is the underpinning of zero trust architecture.

Therefore, it is imperative enterprises ensure correct integration with a trusted IdP provider.

The requesting entity’s identity and profile are considered based on granular policy implemented in Element 7. For example:

**A verified user with the correct profile values**

- would need access to SAP;
- would not need access to an IoT sensor; and
- could need access to YouTube.

**Whereas a verified IoT/OT device with correct profile values**

- would need access to an IoT engine; and
- would not need access to YouTube.

**In addition, a verified workload**

- would need to access a comparable workload; and
- could need access to the internet.

In this simplified example, access policies can be ascertained solely by differentiating the type of initiator. Subsequently, identity can be further assessed and enriched with context, e.g., a valid user logging in from a company device, to deliver a more complete statement of identity (see Element 2).

At this point, authentication moves from simply a contextual yes/no into an authorization stage, where values related to the authenticated identity such as role, responsibility, location, etc., can be used to further validate the initiator.

By combining these values, identity control becomes quite powerful and each identity should be unique at the moment of authorization (re-assessment will be discussed in Element 4 with dynamic risk scoring).

## Technology & Architecture Considerations

A large hurdle enterprises face when getting started with zero trust is technical debt or environmental complexity. An enterprise may have laudable goals, but when it comes time to execute, these factors can obscure the starting point.

Luckily, most organizations already have the baseline technology in place to begin this journey in the form of an IdP with the context of an enterprise IAM.

By reviewing the types of users and role definitions within an IdP platform, IT admins can create an initial sketch of different roles within an organization.

This is not to say that a zero trust identity is solely a value delivered by an IdP. As Figure 16 outlines, identity should consider multiple values by both asking who the entity is and also evaluating the profile of the entity. Nevertheless, an IdP platform should be every enterprise's first step along the zero trust journey. After all, with least privilege, nothing happens without validating identity.

### Pro Tip:

Leverage HR departments or the organizational structure to define your first round zero trust differentiation/segmentation, e.g., finance versus IT users.

An effective zero trust system requires a variety of technical features to accomplish the security checks needed for identity verification. First and foremost, a zero trust solution must have established integration with the enterprise IdP provider. This allows the zero trust solution to not only consume the output of a user identity verification but also receive updates on changes to that verification.

IdP integrations mean the zero trust solution is not the store of identity, but rather where validation and verification happen against a set of predefined controls.

There are four common controls implemented in the IdP platform:

- Two-factor authentication (2FA) such as a card and PIN
- Multifactor authentication (MFA) such as a username, password, and token
- Strong authentication
- Passwordless authentication

Note: Zscaler recommends a minimum of MFA be used to validate users.

Whatever the method, a zero trust solution should consume IdP-provided values of identity, certificates, and shared details including departments, groups, organizational units (OUs), email addresses, etc. The limit on identity values used and the values they contain should be set by the customer. What is key, though, is that these identity values allow for differentiation among users. Figure 17 shows some common examples of identity values that allow for differentiation.

Identity Value	Explanation
Email Address	Normally a unique value per user
Department	Allowing group access under a team value
Employee ID	Normally a unique value per user
Employee Geo	Identifying the region where the employee is employed

Figure 17: A possible set of identity values and their explanations.

This element of the zero trust process is dependent on the functionality of the IdP, including how identity is determined, managed, organized, and updated. As such, the level of identity differentiation will be unique to each company and should commonly be tied to roles as defined by HR.

For workloads and IoT/OT devices, architecting identity verification is quite different and varies widely depending on the deployments. The basic level of categorization will come from the underlying architecture, e.g., “Manufacturing Site A” and “Machine A.”

Additionally, each workload and/or IoT/OT service has a unique set of communication methods, such as destination application requests, unique headers, or other content, as part of the traffic flow that allow for device classification.

Workload and IoT/OT identity verification using site architecture is generally based on network settings and defined trust zones within a network. In other words, “Manufacturing Site A” will have different trust settings than “Manufacturing Site B.”

Further granular identity assessments are possible depending on the tools and machines in use. However, it’s best for enterprises to begin categorization with data and risk classification systems unique to each company.

Note: If the control assessment of Identity and Context cannot be met, the access must default, as outlined in Figure 11, to a Conditional Block policy.

# How does the Zero Trust Exchange accomplish this?

For users, this stage of the zero trust journey begins with robust APIs and standards-based integrations with an IAM, IdP, and other vendors. These enable the ingestion of user identity attributes via Security Assertion Markup Language (SAML), System for Cross-domain Identity Management (SCIM), or Open Authorization (OAuth).

Identity integration within Zscaler is implemented in various categories. This allows enterprises to integrate identity values depending on the need to assess levels of trust. For example, users may not be authenticated, but are all within the bounds of a location, which has a distinct control policy. Zscaler incorporates user and location identity values in the Zero Trust Exchange.

## User Identity

The Zero Trust Exchange has deep integrations with the following [IdP partners](#) (as of August 2022):

- Microsoft Azure AD (SAML & SCIM)
- Okta (SAML & SCIM)
- PingFederate (SAML & SCIM)
- SailPoint (SAML)

Zscaler is also able to integrate with other common IdP providers who authenticate and share authentication values via SAML, including:

- OneLogin
- Google
- Microsoft ADFS
- CA Single Sign-On

These integrations collect common individual attributes an enterprise would provide to differentiate a user, including those mentioned in Figure 17 like department, groups, email address, etc.

SAML Attribute	Example SAML Output
Group_wntynvy-OneLogin	Wntynovy_admin,Wntynovy_domainuser; Wntynovy_user
Department_wntynvy-OneLogin	Employee; Sydney_office; remote
Name_wntynvy-OneLogin	Nathan Howe
Email_wntynvy-OneLogin	Nate_at_wntynvy.com
Geo_wntynvy-OneLogin	Europe; APJ

Figure 18: Example of different identity values taken from a production Zscaler platform.

Zscaler then combines identity data with additional device profile information, sometimes via APIs from other third-party systems like endpoint detection and response (EDR) vendors, to understand the holistic identity of the user.

Understanding an authorized user under various device-based circumstances also allows an enterprise to deliver access controls for users at differing levels of risk. Employees connecting from company-trusted devices should normally have a higher level of access than employees connecting from personal devices, for example. This contextual assessment is outlined in Element 2.

## Location Identity

Circumstances may arise where enterprises need to deliver protection and control for users, workloads, and other miscellaneous devices, but are unable to differentiate between them. A common example would be a shared campus network that contains users, printers, and other devices, where the enterprise may not want to validate any of these initiators. This identity is by no means granular. Quite the opposite, in fact. However, by leveraging this location identity, the enterprise can define policies solely for this location.

## For IoT/OT Devices and Workloads

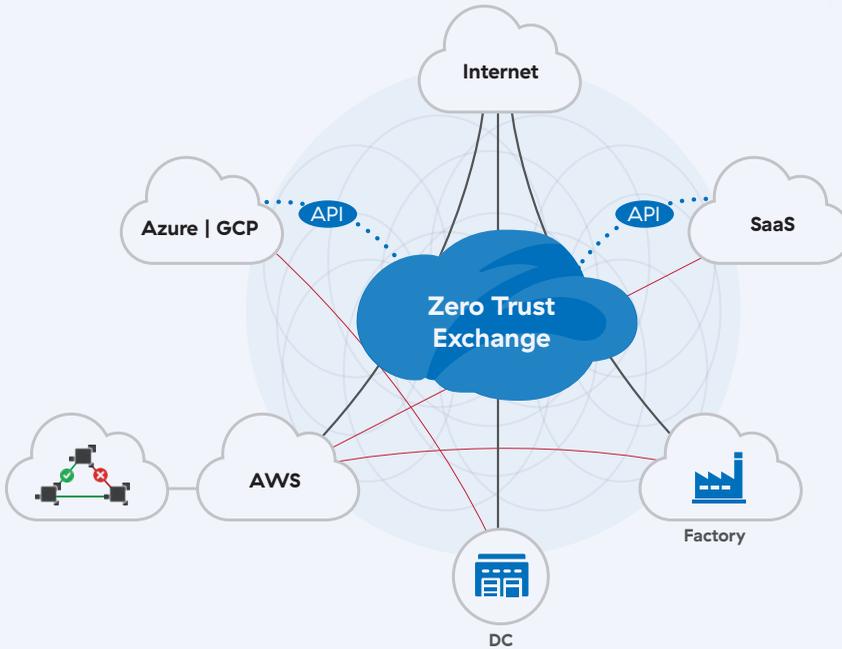
IoT/OT devices and workloads that cannot perform authentication against an IdP require alternative methods to validate the connection source. For example, Zscaler employs unique technology that fingerprints and classifies IoT/OT devices based on their behavior, e.g., a printer will act differently than a camera. In most cases, these devices will be outliers on a device validation path and trust should be determined based on their location.

Workloads may undergo a similar fingerprinting through Zscaler's identity-based segmentation capabilities. Here, an agent can be installed on a workload to ascertain its identity based on a set of characteristics, process functions, permissions, and other local device information.

An example is a customer-hosted API that must communicate with an AWS workload. No identity can be validated against an IdP, nor is it a managed or an unmanaged device. Therefore, leveraging its location is key to establishing its identity.

Workload or IoT/OT communication origins allow an enterprise to architect sites, zones, or "bubbles" that can be considered an identifier. An example would entail an enterprise isolating all "file share servers" within a bubble reserved solely for file share servers. This allows the enterprise to determine identity based on functionality.

Associating location-based values to devices and workloads allows for varying conditions to dictate the identity and ultimately enable the Control and Enforce functions of the zero trust platform.



- **Secure Workload-to-Internet Access**
- **Secure Workload Communication**  
Cloud-to-Cloud, Cloud-to-DC
- **Secure Cloud Posture**  
Secure Public Cloud Data  
Secure SaaS Data
- **Secure IoT/OT**  
Secure Public Internet  
Secure Remote Access

Figure 19:  
The Zero Trust Exchange ensures granular controls are applied to IoT/OT devices and workloads.

Zero trust identity at an IoT/OT and workload level is meant to ensure the appropriate initiating workload can communicate with a destination workload only if authorized to do so. Zscaler leverages the following broad workload identities:

- Locations of workloads, defined by customers to differentiate sites, e.g., DCs versus IaaS
- Sub-location details such as a VPC, VLAN, VNET, or even network criteria that identify different sets of location variables
- Workload network criteria such as IP anchors, connection gateways, etc. (This should rarely be used as a sole value of identity. Zscaler best practices recommend not using IP addresses alone as an identity value.)

# Zero Trust Progress Report

At the conclusion of identity validation, the zero trust process will submit the following values for risk assessment, inspection, and ultimately enforcement.



Progress Report 1: Identity is verified for Jane and John Doe.

# Element 2: What is the access context?

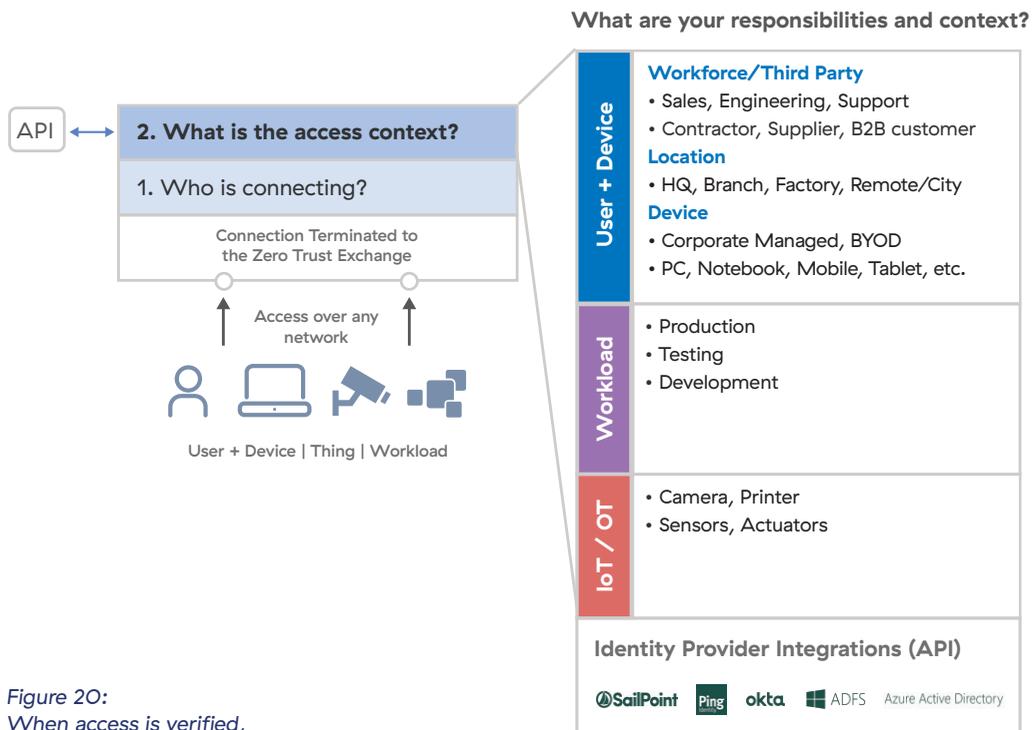


Figure 20:  
When access is verified,  
what additional  
criteria makes up the  
access request?

Confirmation of the initiator (consumed in Element 1) is the first step initially verifying “who” requires access. Identity gives the Zero Trust Exchange an idea of who is connecting, their role and responsibilities, but lacks context around the connection.

Identity is initially considered the ability to deliver a “yes” or “no” binary output based on the initiating entity being authenticated or not. Now we must associate the details of who is connecting with the context of that connection, which allows for additional control of least privilege zero trust. This is made possible by leveraging various identity and profile values at a granular level.

Context in the identity space reveals various insights about the initiator. Continuing the example from Element 1, an employee may identify as Jane Doe. This can be validated by the enterprise IdP. Additional context, however, can be used to further verify their intent, their abilities, and ultimately allow for greater definition of identity.

To demonstrate this context, this time using a workload, the identity may be as simple as two RESTful API processes, let’s call them “device-tickle” and “receive-name.” The context in which these APIs are enabled and employed is what differentiates them from other API calls and processes. Let’s compare these two APIs with contextual differences bolded and underlined:

**device-tickle:** Calls a remote device and uses an HTTP PUT function to tell the remote device “hello.” Note: This is through json (JavaScript Object Notation). This could be used to confirm the remote device is still online.

versus:

**receive-name:** A service that asks the remote device (through an HTTP GET) to share its name. Note: This is in the format xml (eXtensible Markup Language). This call can be used to receive information about remote services.

In these access examples, while both are similar in that they are using the HTTP protocol to execute their function, there are fundamental differences beyond simply the initial identity (name). Given one's ability to prove the variable context, access should be different.

- Context passes verification:
  - device-tickle: called at 9:00 a.m. on Monday, from a trusted DC, through a secure path
- Context fails verification:
  - device-tickle: called at midnight on Sunday, from a remote site

Zero trust controls must allow an enterprise to set granular rules around the context in which device-tickle and receive-name can communicate and access services. This level of contextual granularity can be expanded to many aspects within an enterprise and are not solely related to workloads. The contextual values need to be considered for each enterprise's requirements and included in the Verification of Identity.



### Why is this important?

In the same way that your Netflix login gives you access to Netflix, it is the things about you—age, location, interest, viewing history, etc.—that allows Netflix to recommend shows that will most interest you.

Enforcement of control in zero trust architecture cannot be enabled simply based on who you are. There must be additional applicable understanding and control to set boundaries for access. For example, you may be an authenticated user, but to get access to necessary services you would need verified context to prove several additional aspects:

- The location you are accessing from (country, site, etc.)
- When you requested access (within or outside timeframes)
- How you are accessing (normal patterns vs. exceptions)
- Which device you are connecting from (personal vs. enterprise)

# Technology & Architecture Considerations

Each enterprise will have differing requirements to ensure the correct context is applied within their ecosystem. As such, enterprises should consider the following high-level categories of context:

## Trusted versus untrusted locations

A trusted location should be governed by enterprise-defined conditions that reduce its risk profile. An example would be an R&D lab where all resources are local, isolated, protected, and where an enterprise can ascertain which functional controls can and cannot exist. On the other hand, an untrusted location would be a campus-wide guest network where users connect to the internet with zero access controls.

Note: A location doesn't need to be specific to a site, it can be as broadly defined as a country.

## Defined versus undefined locations

A defined location would be an enterprise office space where users are more trusted than on the open internet. Defined locations may have specific policies applied to them, e.g., user office networks can access the internet, but office server networks cannot. These sorts of network divisions were historically managed by VLANs.

An undefined location, on the other hand, would be anywhere not specifically defined, such as a user's home network.

## Geographic considerations

Defining geographic controls is important not only for security but also for functionality. From a security perspective, user access from specific sanctioned countries should be controlled. From a functional perspective, users should be able to access global resources like google.com in their native language, for instance. Geographic controls can also be used to stop the "impossible traveler" who accesses one service from their device in Sydney followed by an additional service from a location in São Paulo in quick succession.

## Timing bands

The time that a user requests a connection to an application is another contextual attribute zero trust architecture can base policy on. Users accessing certain sites outside of working hours would constitute a different contextual posture versus during business hours.

## Device type

Access to services should vary depending on the device requesting the access. For users, the following context should ultimately define various levels of access:

- Personal vs. enterprise device
- Operating system
- Installed antivirus
- EDR presence

Consider two examples, where the context is very different:

- Jane: On her personal device with an unpatched operating system and no antivirus
- John: On an enterprise device with an up-to-date operating system and EDR running

Similarly, when defining IoT/OT and workload access context, the requesting device plays a larger role in determining access context:

- IoT process on a manufacturing sensor
- OT human-machine interface (HMI)

In both of these examples, contextual details differentiate the access granted.

Of course, additional contextual mapping is possible as an enterprise builds granularity. That said, it's important to maintain the initial idea of zero trust, which states that the underlying network must be considered breached and therefore untrusted. An enterprise will need to operate as though all networks cannot be trusted, with all traffic passing through secure mechanisms over the network.

## How does the Zero Trust Exchange accomplish this?

Given how Zscaler integrates with IdP platforms (outlined in Element 1), it's important to look at various access scenarios to then illustrate how Zscaler provides contextual control. Understanding how to assess these devices and the level of acceptance for each can help enterprises enable various access paths.

Zscaler allows for a wide set of contextual validation integrations. These should be looked at in combination with other tests to deliver a contextual outcome acceptable to the enterprise.

Having these variations of stacked device posture tests allows an enterprise to consume the outcomes of these posture assessments as an additional layer of user context in terms of control and, ultimately, policy.

For user-based devices like desktops, notebooks, tablets, handhelds, etc., we must further differentiate between managed and unmanaged devices:

### Managed Devices

Managed devices are those that can be given a unique identity by an enterprise solution, such as MDM or corporate domain enrollment. These are typically corporate-owned assets where an agent can be installed. Evaluation and differentiation categories are dependent on the customer and deployment, but common contextual tests on managed devices include

- client certificate validation;
- EDR/AV-integration;
- jailbroken or not (for handhelds);
- connected to a known or unknown network; and
- leveraging a trusted or untrusted location.

Note: The Zscaler Client Connector can be installed on these devices to validate these values as part of its zero trust policy without having to call out to external sources for validation. In addition, the Zscaler Client Connector is able to obtain additional insights on context from the IdP-based SAML or SCIM response.

## Unmanaged Devices

These are devices that have no relationship to the organization and include BYOD, third-party, or contractor devices. The ability to assess the status of these devices can be limited, making them immediately less trustworthy than managed devices. But there is a way to differentiate various unmanaged devices, e.g., a contractor working for the organization versus an employee's personal device. Access for unmanaged devices requires different access methods depending on an organization's risk tolerance.

Two common examples:

1. A trusted user connecting from an untrusted, potentially personal device. The personal device must be identified as untrusted and access restricted appropriately.
2. A third-party contractor connecting from an untrusted device. While the user is authorized, the device is not. The contractor's access should thus be limited by leveraging Zscaler's Browser Access and Browser Isolation:
  - The client uses a web browser to access a URL.
  - This URL is actually a CNAME domain that is subsequently redirected to a trusted front-end portal for access, or isolated access, to the app.
  - Web isolation ensures that sensitive data never reaches the untrusted device.
  - This ensures the third party is able to complete the work, but the enterprise is protected.

Note: In most circumstances, the Zscaler Client Connector is not installed on these devices.

It's possible to leverage the Zscaler Client Connector to ensure device validation can differentiate device trust compared to unmanaged devices. That value can then be used to allocate different levels of access. The following are examples of validated users using various devices:

- Validated internal user using a corporate-managed device
- Validated internal user using an untrusted personal device
- Validated external user (not part of primary auth domain) using an untrusted device
- Validated third-party user using an untrusted device

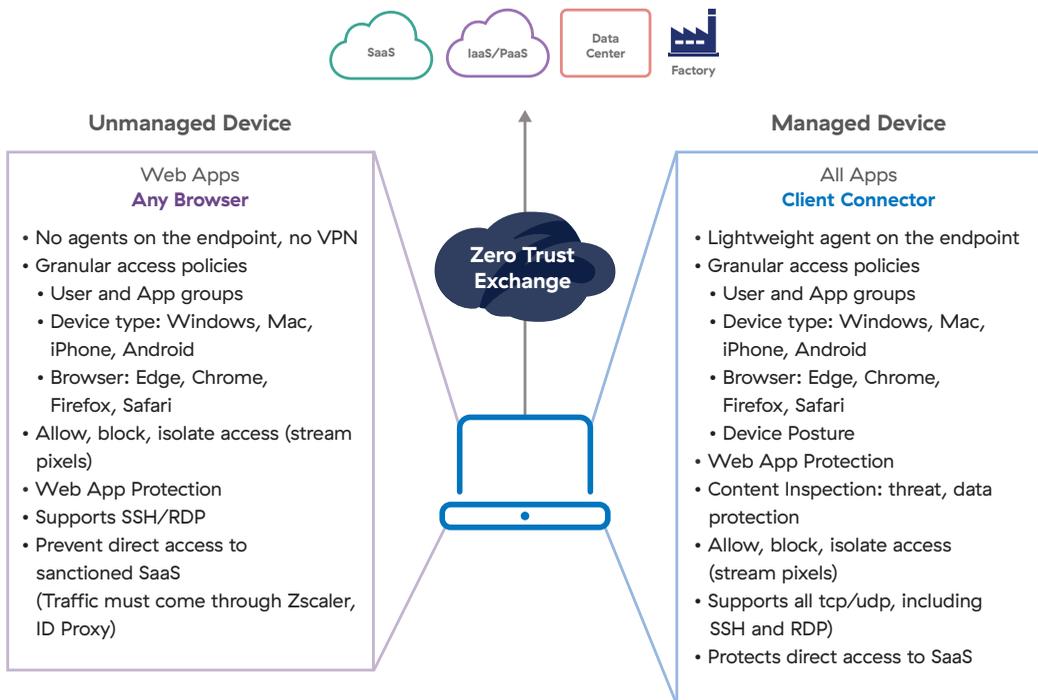


Figure 21: Both unmanaged and managed devices are protected by the Zero Trust Exchange.

Each of these identity verification/validation techniques generates an identity and context outcome to the Application Policy Enforcement engine in Element 7, where the identity can be successfully verified. If it cannot be, or if the policy requirements are not met, the user is blocked from accessing the application.

## Workloads and IoT/OT devices

Zscaler also collects various sets of data to identify workloads. These are broken down into sub-functional groups based on the needs of the enterprise. The Zero Trust Exchange has a unique role in the implementation of control, in that it sits between the initiator and destination.

Ultimately, the goal is to ensure that the appropriate initiating workload can communicate with a destination workload only if authorized to do so. Zscaler assesses the context of workload access based on attributes that include site, location, cloud, and environment level.

For connections between sites, data centers, clouds, the internet, etc., Zscaler is able to consume network criteria, network segments, and IP information to deliver a zero trust policy of access between workloads in various sites.

Connections between workloads within a location, like a VPC, can follow similar network paths and be greatly enhanced through process identity and context validation. This can be achieved and controlled through the deployment of agent software on the various workload systems.

This ability of the Zero Trust Exchange to differentiate access down to a per-request basis of initiator to destination, regardless of the underlying network, allows Zscaler to deliver granular and uniform access controls to workloads.

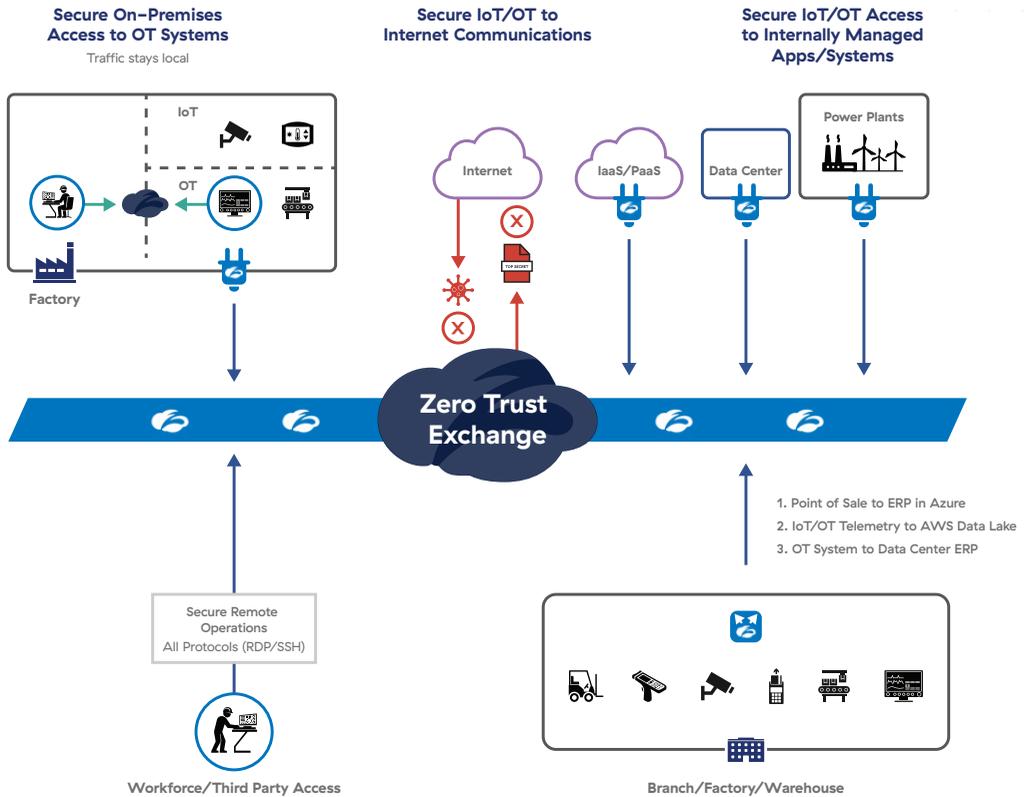


Figure 22: Zero trust for the IoT/OT systems across common access requirements.

As outlined in Element 1, IoT/OT devices cannot be centrally authenticated today. This lack of central control also limits the ability to assign context. As such, Zscaler employs similar techniques for context identification to IoT/OT services as those outlined for workloads, connections, networks, sites, locations, etc.

In these cases, context is often defined by an enterprise rather than automatically scanned or assessed by the Zero Trust Exchange. Customers define various sites or device information that will be consumed as part of the identity and context verification.

IoT/OT services and workloads may undergo a similar fingerprinting through Zscaler's identity-based segmentation capabilities. Here, an agent can be installed on an initiator to ascertain its identity based on a set of characteristics, process functions, permissions, and other local device information.

# Zero Trust Progress Report

The image displays two mobile device screens side-by-side, each showing a Zero Trust progress report. The left screen is for Jane Doe accessing Application X, and the right screen is for John Doe accessing Application Y. Both screens feature a header with user and application icons, a central progress bar with five segments (purple, red, red, light red, blue), and a bottom section with verification details and a large call-to-action button.

**Jane Doe**      **Application X**

**John Doe**      **Application Y**

**Verify Access Conditions**  
**What is the user profile:**  
Untrusted location, using personal device, browser access

**Verify Identity**  
**User Identity and Device:** Jane Doe, 2FA, consultant

**Verify Access Conditions**  
**What is the user profile:**  
Private location, using enterprise device, Client Connector

**Verify Identity**  
**User Identity and Device:** John Doe, MFA, internal

**Jane Doe wants to access Application X**

**John Doe wants to access Application Y**

Progress Report 2: User context has now been added to the access flow.

# Element 3: Where is the connection going?

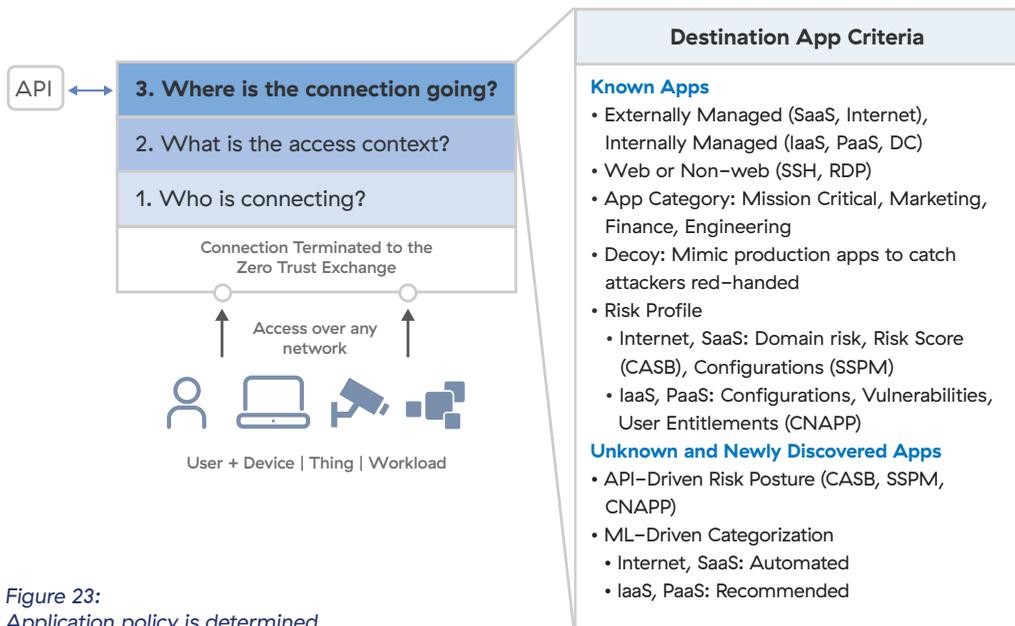


Figure 23:  
Application policy is determined based on a set of criteria.

As discussed, the first element of the Verify process concludes with the initial identity assessment. The next element requires understanding the resource being requested: the application. The app needs to be identified, yes, but its function, location, known risks or issues, and relation to the identity of the access requester must also be evaluated. The app's condition must be understood at a high level. Examples of important considerations include whether the app is known or not, and whether the app is publicly available on the internet.

These conditions will determine how applications are submitted to the Control and Enforce phases of the zero trust process.

Determining which initiator can connect to which destination service is ultimately an outcome of the Verify and Control phases of a zero trust solution. Zero trust services are not firewalls, which means they are neither pass-through nor static. Therefore, the implemented policy must be more than a simple definition.



### Why is this important?

Traditional network controls force all traffic to pass through the same set of controls, regardless of the application type, location, function, etc. Firewalls are famously network-centric and attempt to add application-layer controls by layering them on top of their network function.

In determining why this is important, one must recall how legacy IT controls are implemented statically based on network controls—for example, using IP addresses and TCP port values to identify the service. This is not only limiting and subject to misconfigurations, but also inefficient to set up and maintain.

Take two common apps you may need to access: one internal (like an ERP platform) and one external (like YouTube). These apps have substantial differences in function, form, location, etc.

With a firewall, both apps are treated the same. Controls are applied universally until the path is selected, a decision that typically happens post-control and is reliant on the network (see Figure 24).

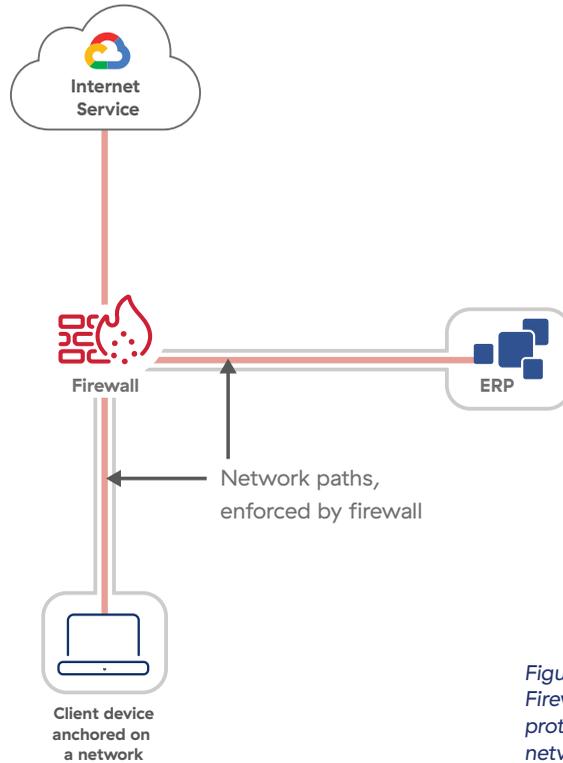


Figure 24:  
Firewall-based  
protection, limited by  
network layer controls.

Firewall-based architecture controls are constructed at layer 3 (network) of the OSI model. As such, they are not natively able to interpret beyond an IP address. This means any attempt to understand content beyond the IP-to-IP stateful control, such as the identity of individual users brokered from an IdP (SAML/SCIM/etc.), requires additional bolted-on functions or infrastructure, not only to manage identity values but also to associate any traffic with this identity and enforce required controls for these sessions.

This is impactful since cloud-hosted applications leverage IdP-based authentication and authorization, so features like single sign-on (SSO) are seamless when a user logs into a corporate Salesforce account or an internal SAP ecosystem.

Not understanding these authentication and authorization outcomes results in two distinct, negative impacts when using firewalls:

1. Users must authenticate twice with two different authorizations.
2. These identity values must be managed in two locations, with two different sets of identity controls to consider.

Conversely, deploying a zero trust solution that is natively based on layer 4–7 proxies allows for inline integration and understanding of identity in relation to users’ access requests. This means that, when access is requested with a true proxy-based zero trust solution, the control focuses on the identity and conditions of the initiator (all the values outlined in Element 1) plus the context of the destination application, rather than solely an IP address. User-to-application segmentation can therefore be achieved not based on cumbersome network controls but rather by identity-aware policies.

This allows a zero trust solution to assess end-to-end (not solely network-based) context and apply controls for more granular categorization and access control.

With a zero trust solution, applications are evaluated individually. The ERP app is recognized as an internal app that should be utilized by few users, while YouTube is recognized as an external app available to anyone. Infrastructure, locations, and IP addresses related to YouTube are easily identifiable and should be actively updated within application context.

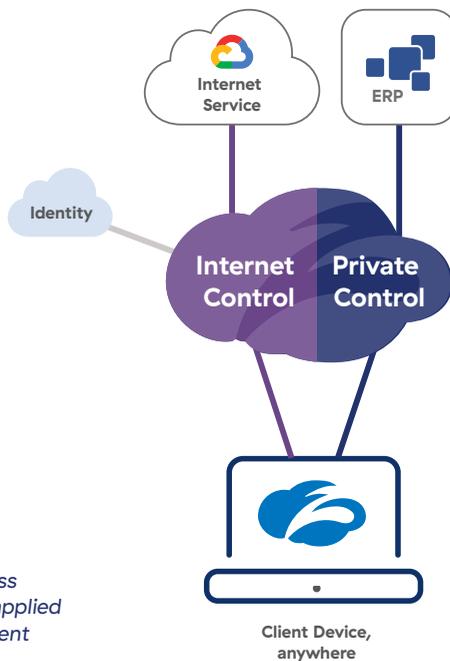


Figure 25:  
Zero trust app access  
policy controls are applied  
regardless of the client  
application location.

Foundationally, all services within a zero trust solution must not be trusted. Trust considerations have substantially shifted due to the dynamic nature of content and applications accessed. Least-privileged access in zero trust delivers multiple benefits to enterprises:

- Applies the correct controls to the correct source
- Obscures protected resources from unauthorized sources, reducing cybersecurity risks
- Reduces waste, e.g., a Linux server isn't allowed to connect to a Windows patch system
- Provides granular visibility and learning of flows per access request, not network IP-to-IP
- Consolidates access based on identity and not on a network, allowing a network's function and infrastructure to be optimized

### Pro Tip:

Defining application segmentation policies can be daunting given the size of enterprises. Below are three steps for beginning the segmentation journey:

1. Determine critical workloads and who should access them, beginning with a specific policy to protect "known-critical" assets if possible. If not, start with step 2.
2. For all other traffic, obtain visibility over access, thus giving visibility and an inventory of apps with a discovery policy.
3. Learn from insights offered by user-to-workload traffic flows and iterate your policies. An example of best practices for optimized policy is outlined in Appendix 1.

Creating application segmentation policies can be greatly simplified with machine learning insights (see Figure 26 below).

### AI radically simplifies app segmentation in large, flat networks

Shrink the internal attack surface from 20,000 employees → 50 employees

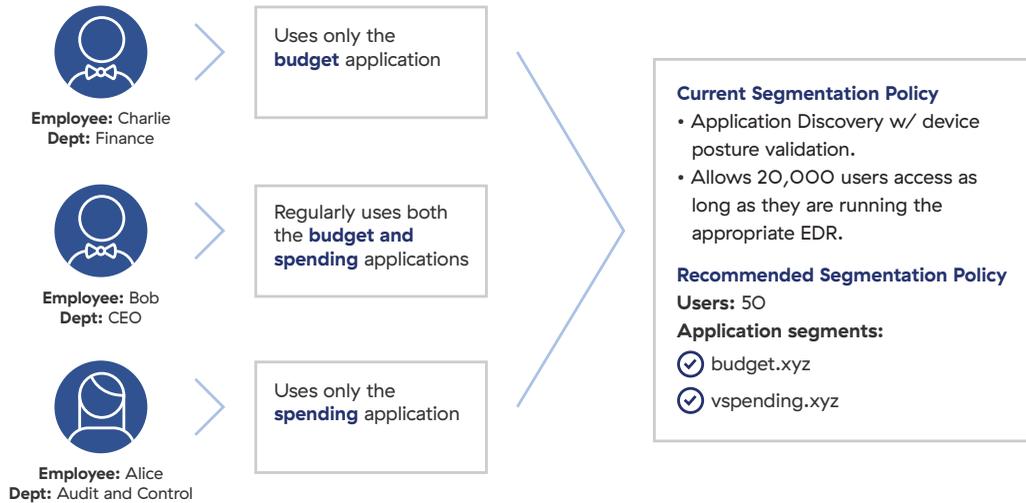


Figure 26: Machine learning radically simplifies app segmentation in large, flat networks.

# Technology & Architecture Considerations

The process of determining the application category and policy begins after validating the initiator's identity. Based on the initiator's request, applications must be differentiated between types:

- External or internal apps
- Known apps belonging to a predetermined category
- Unknown apps

Connectivity is not the goal of app determination. Rather, it's the implementation of rules to decide what conditions will be considered within the Control and Enforce phases.

## External or internal apps

- **An external app** is one that is consumable from anywhere on the internet and has some sort of inbound listening service that allows anyone on the internet to attempt to connect to it. These are apps that exist on the open internet like google.com or salesforce.com.
- **Internal apps** are those hosted by the enterprise in their data center or in an IaaS/PaaS environment that have an inbound listener, but are generally privately hosted behind network layer controls like firewalls and connected to internal trusted network paths. These apps exist in internal address spaces, e.g., server1.local, or on a private IP (RFC-1918) space.

## Known apps belonging to a predetermined category

These are applications that the zero trust system already knows and can be classified into one of three categories:

- **Known good:** Applications that are documented, assessed, and understood, e.g., salesforce.com
- **Known bad:** Applications that are documented, reviewed, and determined to be malicious, e.g., illicit dark web marketplaces
- **Known risky:** Applications that are documented, reviewed, and are possibly risky depending on who accesses them, e.g., InfoSec websites

## Unknown apps

These are applications that the zero trust system has newly discovered and has not yet categorized. These apps should be considered untrustworthy and risky until proven otherwise. This ensures Control and Enforce policies scrutinize these apps at the highest level.

An app's risk to the enterprise must be identified and categorized appropriately.

If unknown apps are external, i.e., consumable on the open internet, the zero trust solution should be able to quickly assess their risk level. This assessment concludes with a categorization of the site based on function, such as a video streaming site versus a sales tool.

Internal apps must be flagged as newly identified, allowing the enterprise to determine which segment, app definition, policy, etc., best describe them.

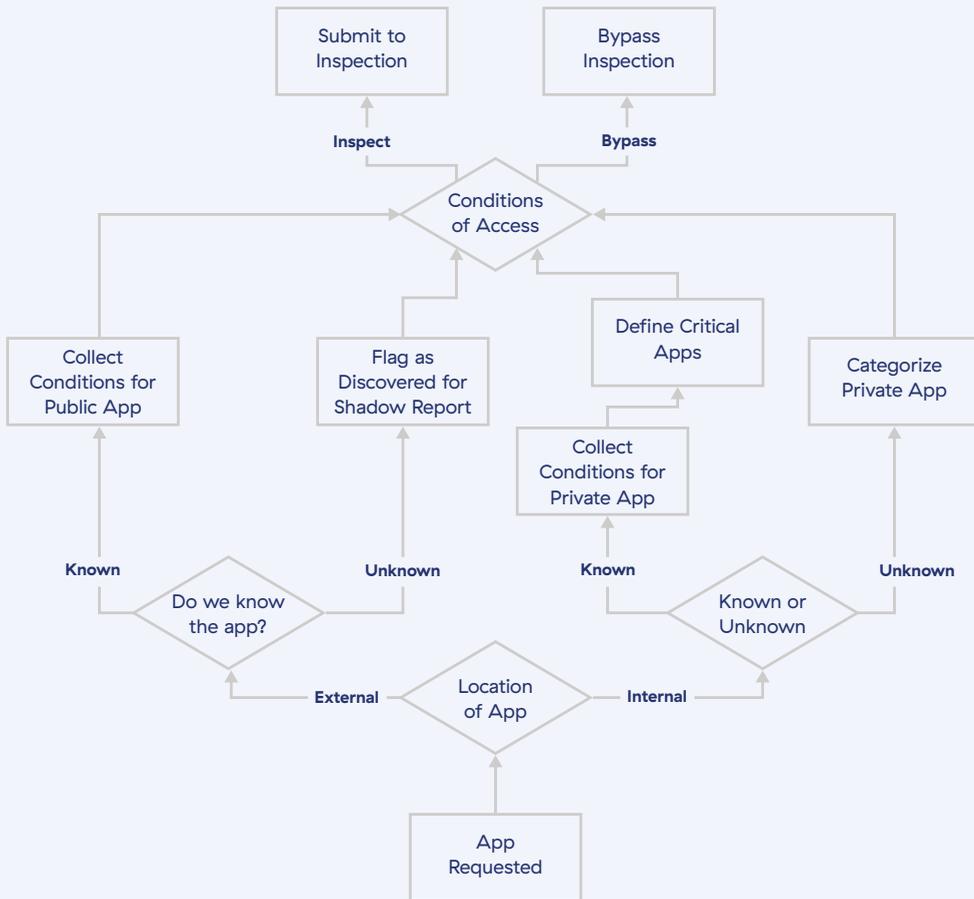


Figure 27: Flow diagram depicts application categorization and policy enforcement process culminating in an inspection decision.

There is further differentiation between application types, including IT and IoT/OT workloads, web and non-web apps, decoy apps, and critical apps.

## IT vs. IoT/OT workloads

Both sets of apps generally rely on similar infrastructure and technology: connected devices communicating on a shared network. However, there are major differences:

- Information technology (IT) apps deal solely with the processing of information for human users, e.g., email or video streams.
- IoT apps generally collect data from IP-enabled things and forward it to external IoT platforms for analysis.
- Operational technology (OT) apps control physical devices, e.g., industrial or access control systems.

OT is unique in that the hardware and software are designed to monitor and control specific physical environments, such as heating a space or slicing food. Typically, OT controls follow a standardized set of OT security controls depending on their function (e.g., [IEC 62443](#) or the [Purdue Model](#)).

OT services can manage industrial control systems (ICS) and supervisory control and data acquisition (SCADA), which importantly often require human oversight. IT-based technology solutions like zero trust are able to provide [privileged remote access](#) connectivity to OT platforms. This allows for fully isolated, clientless RDP/SSH as well as secure file/data transfers.

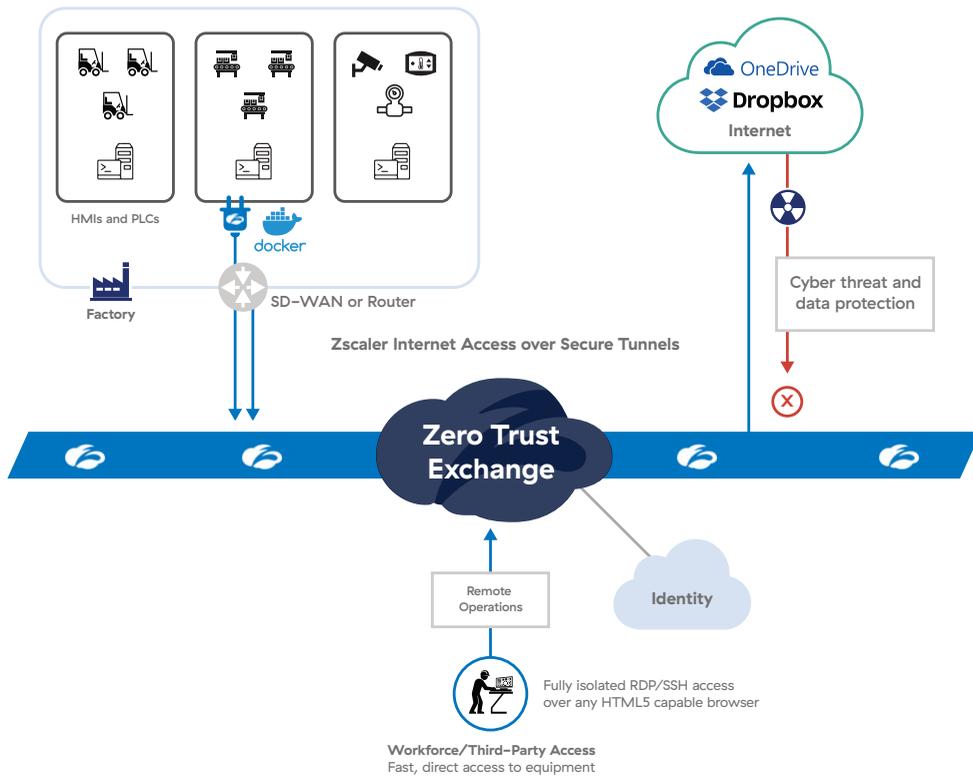


Figure 28: Privileged remote access for OT systems.

## Web vs. non-web apps

While web-based apps are the majority of traffic (over 80%), they are not the only traffic requiring categorization. Non-web apps might include categories enterprises map to certain functions, like “All file servers” as a group that contains all IP addresses, FQDNs, and domain names for all servers that host file servers running on port 445 (SMB).

## Decoy apps

Policy definitions should include the ability to define alternate destinations for app access. This allows enterprises to define honeypot-like services that help identify and mitigate threats posed by compromised users.

## Critical app definition

Not all applications, segments, or groups are created equal. Each enterprise must differentiate between app types based on its business and priorities. This normally involves risk and data classification requirements and/or enterprise “key assets.” Differentiating these in policy is critical to subsequently defining access, roles, controls, etc.

Critical applications could be considered core to the business function, such as an ERP system for a logistics company or a CAD system for a design company. Ideally, these apps receive the most restrictive set of rules and controls, allowing only necessary user access.

Critical apps must be differentiated in policy and access control from less critical apps, where access is more open. Enterprises must clearly define and manage these apps. Ideally, this would be based on a classification system taken from standards like SOC 2, PCI, or similar. Enterprises should consult internal security risk and compliance teams to align specific classification requirements.

These destination values, coupled with Element 1 outputs, allow a zero trust solution to make a definitive policy decision and apply enforcement within Element 7 in accordance with enterprise requirements.

Note: If the App Policy cannot be met, access must default to a Conditional Block Policy.

# How does the Zero Trust Exchange accomplish this?

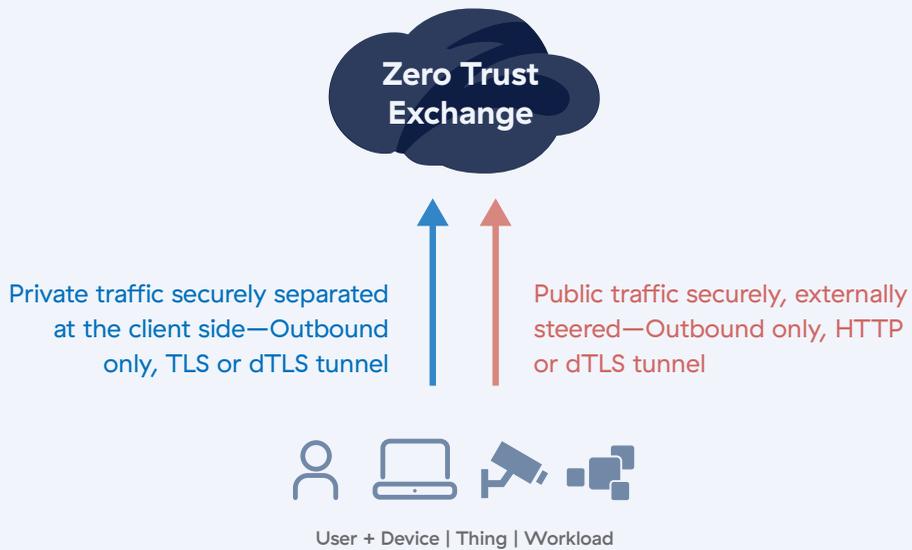


Figure 29: External and internal traffic are split at the source.

The requested application is assessed only after Zscaler has validated the initiator in Element 1. Zscaler’s assessment involves multiple phases to ensure three key points:

## 1. External and internal traffic are separated at the source

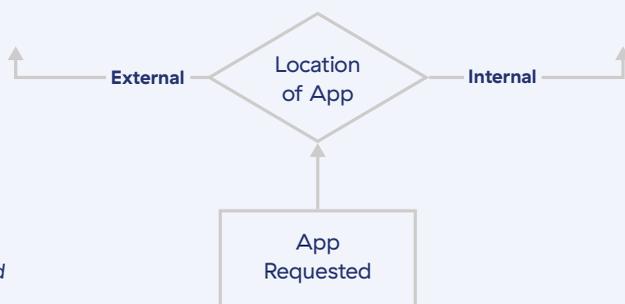


Figure 30:  
Determination of the  
path and controls based  
on traffic type.

Rather than sending all traffic to a distant edge service to determine external and internal paths, as with typical network-based connections like VPNs, the Zscaler Client Connector is able to intelligently understand the type of application (external or internal) and steer it over the correct tunnel to the correct security edge service, as in the following instances:

- YouTube is an external internet app and must have its controls implemented at the internet control layer.
- erp.company.local is an internal app and requires not only a different set of access controls but also separate access paths.

By intelligently breaking off traffic at the client layer, Zscaler delivers two distinct advantages:

1. End-user Experience – Traffic is evaluated according to its own, unique access path. The lack of dependency on networks for control allows for effective and best-path selection to be determined per app.
2. Simplification – Policy enforcement is performed where required, inline between the initiator and the destination. It is not centralized with lots of complex routes or network interconnections needing to be managed.

## 2. Application categorization

Figure 31:  
Process to decide  
if the app has been  
seen before.



Once an application request reaches the correct enforcement point, the app’s actual function and category are evaluated, categorized, and if necessary, updated. This ensures that, once the Control and Enforce phases process the traffic, the correct control is applied to the correct access request.

The first step is to determine if the application is known or unknown, asking whether the Zero Trust Exchange has already categorized it or not.

If the application is known, it means that it is part of an existing and predefined app category. This is not always simple, however, and requires dynamic app categorization. Take, for example, internet-based services that scale with demand. Updating control policies in network infrastructure means adding new IP addresses to the rules.

In dynamic app categorization, any additional application information can automatically be added to the app and/or [cloud app category](#). Zscaler maintains predefined and updated categories, such as the one-click integration with Microsoft 365, to address these requirements.

Figure 32:  
Application  
example  
with multiple  
associated  
addresses  
(and services).

```

; ; ANSWER SECTION:
co2br.prod.zpath.net. 576 IN CNAME co2br.gslb.prod.zpath.net.
co2br.gslb.prod.zpath.net. 45 IN CNAME fra4.co2br.gslb.prod.zpath.net.
fra4.co2br.gslb.prod.zpath.net. 4 IN A 165.225.73.254
fra4.co2br.gslb.prod.zpath.net. 4 IN A 165.225.73.252
fra4.co2br.gslb.prod.zpath.net. 4 IN A 165.225.25.209
fra4.co2br.gslb.prod.zpath.net. 4 IN A 165.225.25.210
fra4.co2br.gslb.prod.zpath.net. 4 IN A 165.225.25.208
fra4.co2br.gslb.prod.zpath.net. 4 IN A 165.225.73.251
fra4.co2br.gslb.prod.zpath.net. 4 IN A 165.225.73.253
fra4.co2br.gslb.prod.zpath.net. 4 IN A 165.225.25.207
  
```

## Zscaler provides customers multiple ways to categorize applications:

### Predefined

Common apps can be prepopulated with data available on the open internet. This mostly applies to internet-based services.

### Manual Definition

Details of applications are set by the client. While internet-based services are rarely left to customers to define, internal applications are often best defined by the client, especially when restricted.

### Cloud Definition

This includes services consisting of various sets of apps and functions, like YouTube. Zscaler leverages active learning from traffic across the cloud to ensure the latest information is included in the application policy.

### API-Driven Definitions

Leveraging integration with cloud services, Zscaler actively learns from platforms, like Microsoft 365, to deliver the latest app segment definitions.

### Unknown and Newly Discovered

This category is used when initiators request an application unknown to Zscaler or our partners. Identification of the services and applications making up a policy must be dynamically updated. Ensuring that all new functions, services, IPs, hostnames, etc., are automatically updated means that—when an enterprise defines a policy of “block all video streaming,” for example—all new IPs related to YouTube, TikTok, etc., are detected and added to the policy controls.

Zscaler actively communicates optimizations to group categorizations through email and web updates, so it's clear whether apps, categories, or values are changing.

Domain	Current URL Category	Updated URL Category (from 04/24)
getdropbox.com	Professional Services	File Host
datastudio.google.com	Web Search	File Host

Figure 33: Example of URL re-categorization of a newly discovered app.

Zscaler allows any internet-connected individual to assess its current categorizations and suggest changes through its [Site Review assessment tool](#).

If the application is not known to the Zero Trust Exchange or the customer, it must be flagged as newly discovered. Zscaler subjects newly discovered apps to various assessments to understand their function, risk, and other insights. Within Zscaler, the app is categorized and evaluated in the following manner:

1. Assess the app to identify its type, e.g., web app
2. Identify any reputation for the app or domain
3. Assess the content and function of the app
4. Categorize, if possible
5. If not, flag as uncategorized



Figure 34: Process to differentiate unknown from known apps & categorize.

Uncategorized apps receive separate conditions in Element 7 to determine the appropriate actions and outputs.

App categorization by group allows for efficiencies in the size and scope of apps to be taken into account.

The Zscaler Zero Trust Exchange creates and manages apps across various categories, including:

- Collaboration & Online Meetings
- Consumer
- DNS Over HTTPS Services
- Finance
- Healthcare
- Hosting Providers
- Human Resources
- IT Services
- Legal
- Productivity & CRM Tools
- Sales & Marketing
- System & Development

Visit Zscaler Help for a full list of [cloud app categories](#) or [URL categories](#).

Zscaler automatically updates most of its predefined app categories using knowledge collected from the cloud, partners, threat intel, and other sources.

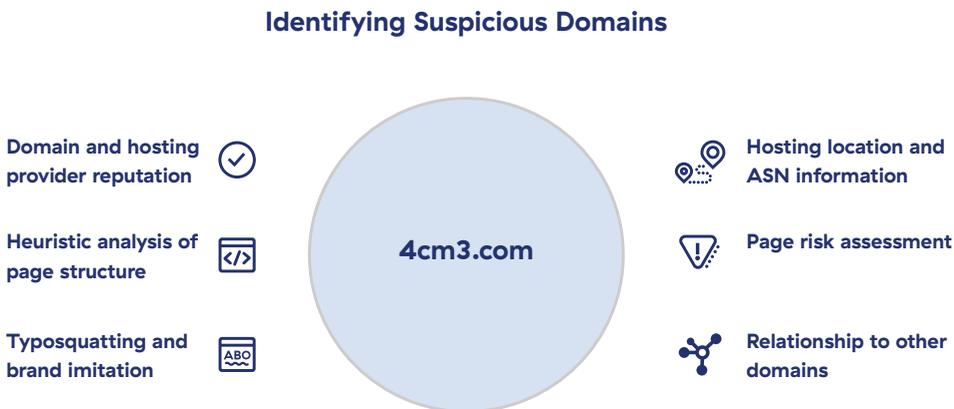


Figure 35: The Zero Trust Exchange uses numerous factors to identify a suspicious domain.

Customers can also manually create and modify their own app segments based on their application classification requirements.

### 3. Application policies based on categories

After an app is categorized, it must be assigned to a policy to ensure traffic destined for the app is correctly controlled. Zscaler implements app categories for various sets of functions relevant to policy enforcement.

Note: A complete overview of access policy is outlined in Element 7.

App Policy is built within Zscaler by applying application categories to various security controls. These categories for internet services include the following examples:

- Web-based (HTTPS) gambling websites should be restricted
- Cloud apps, like Microsoft 365, should be allowed with controls
- Apps with sensitive data where data loss prevention should be deployed, both inline and out-of-band
- Bandwidth-intensive apps where bandwidth controls should be applied

Applications must be correctly defined and categorized to ensure the subsequent Control and Enforce elements are appropriately applied.

## Decoy apps

An entirely unique type of application exists whereby the Zero Trust Exchange will identify and direct a compromised user to a decoy cloud. This decoy cloud will mimic the behavior of an actual internal application while preventing lateral movement and the loss of sensitive data. By observing behavior of the compromised user within the sensitive application, further damage is prevented.

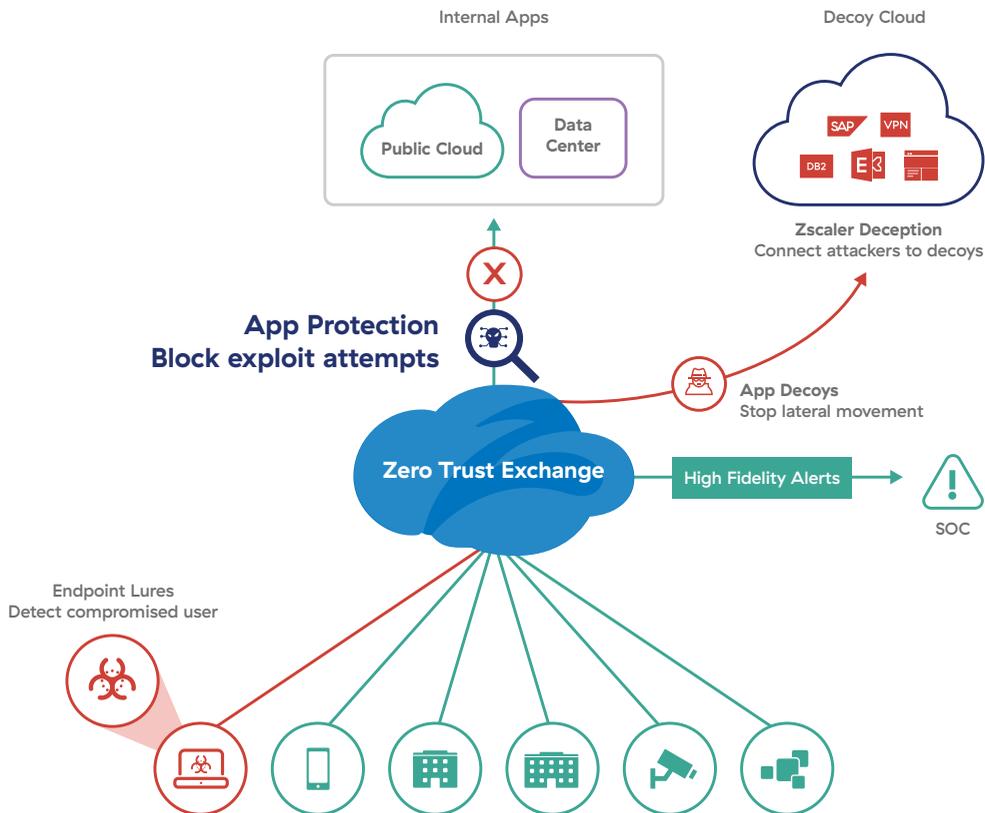


Figure 36: Send attackers to decoy and not live environments.

## Application segmentation

Once application destinations are categorized with appropriate access control policies, it is then necessary to specify which groups of users can access those applications. This allows for user-to-application segmentation independent of network-based controls. These controls are implemented based on access policy rules. These rules define the users and then define which applications or segment groups they can access.

For application-based access control, an enterprise can create an access policy rule for specific application segments or segment groups. Different policies can be used to grant access for individuals or groups of users to individual applications or across a group of applications. The criteria used to create these controls include user profile, device posture, connected network, IdP attributes, and others. These attributes can be used to create segment groups or machine groups.

The creation of access policy rules can seem daunting, especially when moving from a VPN-based solution where such rules were not needed since users were granted wide network access. It's often useful to start the app segmentation journey with no segmentation at all, and leverage the Zero Trust Exchange by applying a \*.\* application policy. While this mimics the level of access provided by the VPN, it has the benefit of removing the attack surface caused by an externally exposed VPN concentrator. Using this as a starting point, the enterprise can go on to create more granular access control policies. Machine learning-based techniques allow Zscaler to recommend access policies based on the actual traffic flows. More details on application segmentation can be found in Appendix 1.

# Zero Trust Progress Report

Identity and context have been verified for both John and Jane, and the next step is identifying where the connection request is headed, as well as the implications of that application connection.

User	Application	Verify where the access is going	Verify Access Conditions	Verify Identity
Jane Doe	Application X	Internal ERP service on port 23 in clear text External new Social Media site on port 443, encrypted	<b>What is the user profile:</b> Untrusted location, using personal device, browser access	<b>User Identity and Device:</b> Jane Doe, 2FA, consultant
John Doe	Application Y	Internal File share service on port 445 External video sharing site on port 443, encrypted	<b>What is the user profile:</b> Private location, using enterprise device, Client Connector	<b>User Identity and Device:</b> John Doe, MFA, internal

**Jane Doe wants to access Application X**

**John Doe wants to access Application Y**

Progress Report 3: App policy is determined for Jane and John.

# Section

# 2

# Control

# Control

Content and Access

- 6. Prevent Data Loss
- 5. Prevent Compromise
- 4. Assess Risk (adaptive control)

There’s no point applying controls, computing power, or even electricity to something that’s outside your mandate to protect. In fact, controls should never be wielded against objects outside your control structure. This is why, in Section 1, we validated who the identity is, what the context of the connection is, and where the connection is going.

In Section 2, it’s important to take decisions of identity, context, and application policy and start applying the first levels of control policy. This requires the zero trust architecture to break initiators’ connections and examine what they’re doing, much like an airport security checkpoint. Section 2 is about understanding risk and looking inside to see what’s going on. Element 4 describes how risk is assessed, while Element 5 and Element 6 discuss how content is inspected for cyberthreats and possible sensitive data exfiltration.

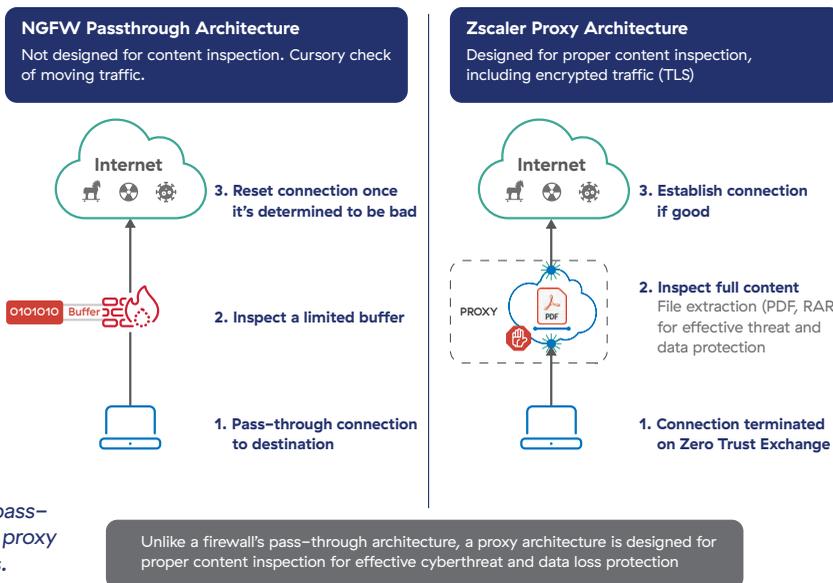


Figure 37: Comparing pass-through and proxy architectures.

It is important to remember that this level of control is difficult to achieve with firewall-based architectures and necessitates the move to a zero trust architecture. Pass-through, stateful controls of legacy firewall deployments require additional layers of services be daisy-chained or bolted on. This is compared to a true inline proxy solution from Zscaler, where additional controls are included directly with no additional layers needed.

# Element 4: Assess Risk (adaptive control)

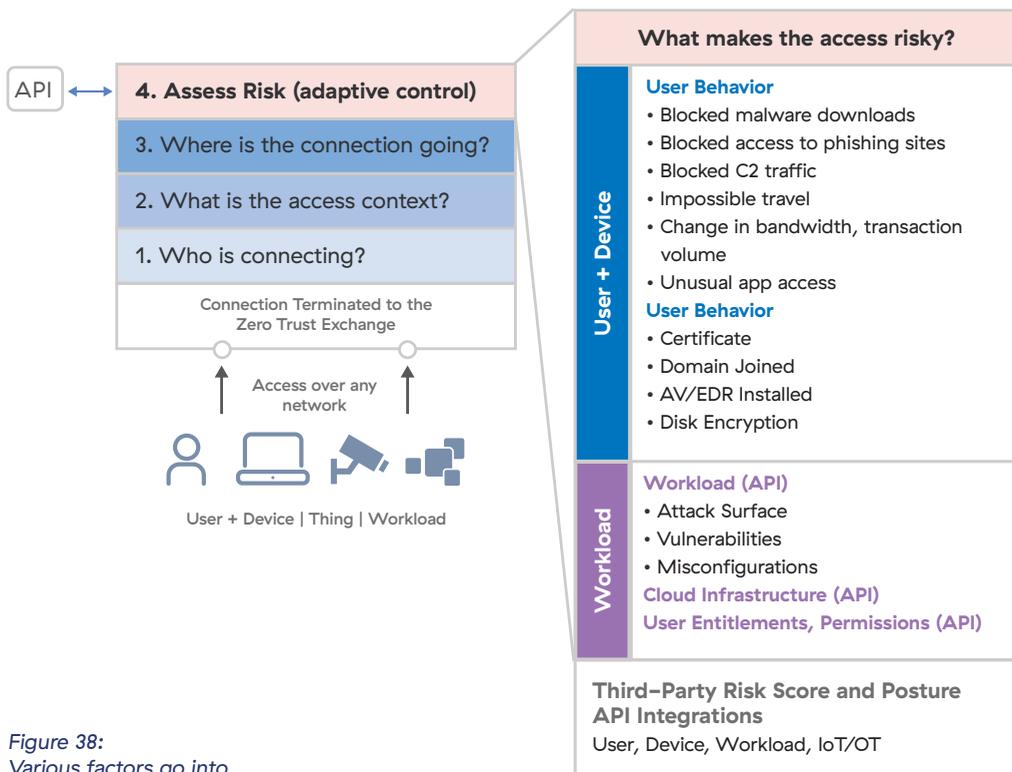


Figure 38:  
Various factors go into the calculation of a dynamic risk score.

In life, we're all judged by the outcome of our last performance. The same goes for zero trust. The previous elements are only as good as the latest assessment. The solution must account for an enterprise's tolerance for risk via a dynamic risk score.

Subsequent and ongoing risk assessments must be dynamically calculated from a set of signals to ensure the score is updated based on new developments. This risk then feeds into a decision engine to determine whether ongoing access should be granted.



### Why is this important?

All values captured in Elements 1–6 must be presented at Element 7, the enforcement layer. In its raw form, this is simply the submission of values to apply policy, enforce controls, and establish connections as allowed.

True zero trust requires continuous evaluation of all values related to the authentication, inspection, and access determinations based on the destination service.

For example, identity verification must be continuously updated in the event the original authentication criteria are no longer met.

Dynamic risk calculation is important for making risk-based access decisions throughout the life of a connection. A marked change in user/device posture or behavior can trigger an update on the access decision in near real time based on updates to a risk score.

# Technology & Architecture Considerations

Calculating risk dynamically requires varying inputs that cannot be consumed in a uniform manner. There must be mechanisms to regularly assess user/device profile values to confirm the status, change, updates, and ultimate risk of the initiating request.

## Pro Tip:

Risk can be complicated to calculate, so start with areas worthy of further investigation, e.g., third parties vs. employees. Then, narrow the category to include third parties with unlimited access compared to those with limited access.

Understanding user behavior is critical to distinguishing between risky and benign actions. This analysis learns patterns of normal behavior to then determine when behavior is anomalous. Analyzing these patterns and evaluating them against company policy leads to better access request determinations.

More simplistic methods of risk calculation focus on measuring rates of undesirable traffic or blocked access requests generated by a user.

Collection and collation of risk inputs also can't be limited by an initiator's location. While these methods give some indication of risk, they are neither truly dynamic nor based on a sufficiently broad set of input criteria. Additionally, calculating accurate risk scores isn't workable as an add-on service.

Dynamic risk scoring must be a fundamental feature of a zero trust solution, able to scale with an enterprise's ideas of acceptable risk. Values must be collected regardless of whether the initiator is connected at home, a coffee shop, in the office, or elsewhere. Zero trust solution providers must deliver global, highly-available, scalable, and network-agnostic solutions that offer consistent policy decisions and a seamless user experience while mitigating risk.

Just as behaviors vary among user identities, workloads must also be evaluated relative to their known and comparatively static activity; an SQL client should talk to an SQL server, but rarely if ever should it communicate with an unrecognized server.

Independent of the request initiator's identity, the outcomes of these assessments are used to create a risk score, which is then sent to the application policy decision engine described in Element 7. It's at this point where the decision of which action to take, based on the risk score calculated from user posture, device posture, and anomalous behavior is made. Customers must also decide on the frequency with which this determination must be made when risk scores are dynamically evaluated.

Third-party solutions can provide additional insight into user and workload risk assessments. Those garnered from EDR, SIEM, or SOAR services may add context for improved determinations.

Note: If the control assessment of Risk Score cannot be met, the access should default, as outlined in Figure 11, to a Conditional Block policy.

# How does the Zero Trust Exchange accomplish this?

The Zero Trust Exchange includes multiple mechanisms for identifying issues and ultimately calculating company-wide risk scores. This allows enterprises to view and investigate risk using

- risk score by authenticated user (showing which known and authenticated users are considered risky);
- comparison of the enterprise risk against its industry vertical;
- risk distribution through an organization (e.g., is IT riskier than sales?);
- behavior of the riskiest users;
- locations with the highest risk.

These values are dynamically identified and delivered to the policy enforcement stage discussed in Element 7, allowing enterprises to regulate access based on the latest, dynamically collected scores.

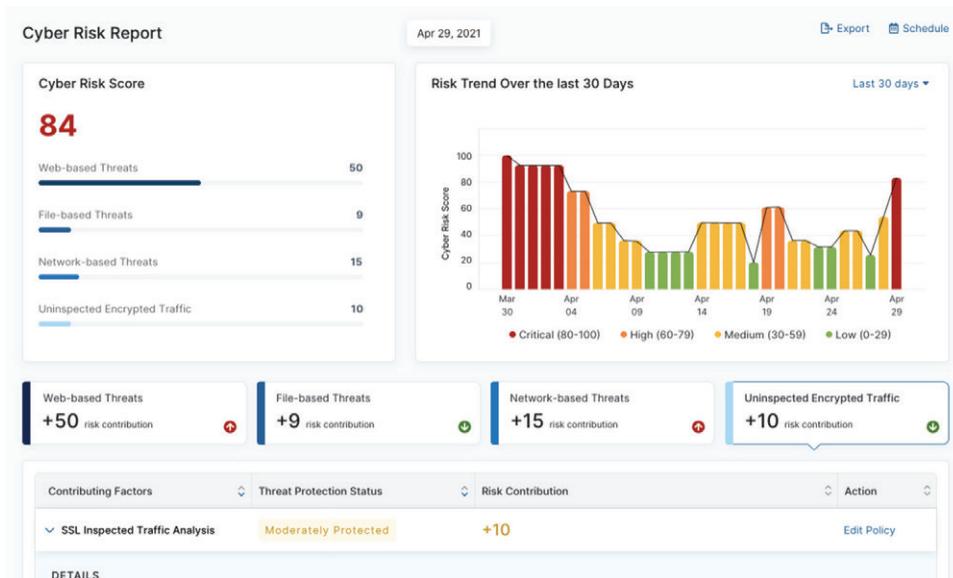


Figure 39: Zero Trust Exchange Cyber Risk Assessment.

Zscaler’s risk scoring is informed by a proprietary algorithm that takes profile, posture, and behavior into account before assigning a value. These risk values and their outputs vary based on the entity making the resource request, be it a user, IoT/OT device, or workload.

The goal of risk evaluation is to allow customers to apply organizational, locational, departmental, and user-level risk telemetry to policy enforcement. This, in turn, allows for greater control and visibility of overall risk assessment and management strategy.

Beyond company-wide risk scoring, there is also user-based risk scoring. There are three key categories of user-based risk evaluation, which include pre-compromise behavior, post-compromise behavior, and suspicious and anomalous behavior.



Figure 40: An example user risk score measures against factors including overall risk and comparison with similar employees and peers.

The Zero Trust Exchange actively collects information to support risk assessment scores from the individual customer’s set of configurations, uses, and insights—for example, looking both at the number of malicious callbacks from a client during a time frame along with global inputs from the Zscaler cloud.

An example outcome of a user risk score analysis that would result in a particularly high risk score looks like this:

### Events that contributed to the score over 7 days

#### Yesterday

- Suspicious modification of Sensitive Groups  
Oct 26, 2021, 2:21 PM |  Suspicious
- Malicious file Blocked  
Oct 26, 2021, 2:21 PM |  Malware
- Detected possible botnet command and control traffic  
Oct 26, 2021, 2:21 PM |  Infected
- Violates Compliance Category  
Oct 26, 2021, 2:21 PM |  Suspicious

Figure 41:  
An example of events that will influence how a user’s risk level is calculated.

Zscaler customers can quickly create views of user-centric and [company-wide risk](#). This visibility, along with insights from [Zscaler’s ThreatLabZ team](#) and the global cloud effect, allows for accurate categorizations of user behavior, which are then applied to policies to ensure dynamical control.

Workloads generally have a limited set of risk identifiers for defining how one differs from another. Thus, they should be considered based on the location from which they attempt to initiate a session. Defined as such, workload risk scores are a function of a location’s sensitivity combined with tendency toward anomalous behavior. For example, a protected file share workload should not have access to Netflix, with any deviation being cause for a change in the site’s risk evaluation.

Given the lack of unifying identity solutions for IoT/OT devices and the fact that identity is static, the riskiness of any “thing” is calculated by Zscaler using device traffic flow. As such, traffic flow is broken into two categories, outlined in Element 5 and Element 6.

## Cloud-native application protection (CNAPP)

When delivering services across many IaaS, PaaS, and SaaS offerings in addition to leveraging microservices and serverless architectures, siloed on-premises security solutions can't scale fast enough to secure mission-critical cloud applications anymore.

The Zscaler Zero Trust Exchange offers a cloud-native application protection platform (CNAPP) that takes an approach to cloud-native application security with an agentless solution that correlates multiple security engines to prioritize hidden risks caused by misconfigurations, threats, and vulnerabilities across the entire cloud stack, reducing cost, complexity, and cross-team friction.

These security engines operate out-of-band to provide another level of risk assessment, focusing on ensuring that cloud entitlements and security posture conform to enterprise requirements.

# Zero Trust Progress Report

After a dynamic risk assessment, risk values are included with original identity outputs to round out the view of the requesting entities, John and Jane Doe. The zero trust process will consume this verified identity as part of its policy implementation.

The image displays two side-by-side progress report cards for Jane Doe and John Doe. Each card is a rounded rectangle with a light blue border. At the top of each card, there are two circular icons: a person icon for the user and a document/chart icon for the application. Below the icons are the names 'Jane Doe' and 'Application X' for the left card, and 'John Doe' and 'Application Y' for the right card. The main body of each card is divided into several horizontal sections with colored backgrounds: a light purple header, a red section, a light red section, a light orange section, a blue section, a light blue section, and a white section. The 'Compute Dynamic Risk Score' section in the light orange background lists 'User Posture', 'Device Posture', and 'Behavior' with their respective values and a final 'Risk Score'. The 'Verify where the access is going' section in the blue background lists internal and external services. The 'Verify Access Conditions' section in the light blue background lists user profile details. The 'Verify Identity' section in the white background lists the user's identity and device. At the bottom of each card, there is a circular icon of the user and a text statement: 'Jane Doe wants to access Application X' and 'John Doe wants to access Application Y'.

User	Application	User Posture	Device Posture	Behavior	Risk Score
Jane Doe	Application X	Untrusted network	AV only	Repeated login attempts	High
John Doe	Application Y	Trusted network	AV/EDR, Disk Encrypted	Normal	Low

Progress Report 4: The calculation of risk for Jane is quite different than for John.

# Element 5: Prevent Compromise

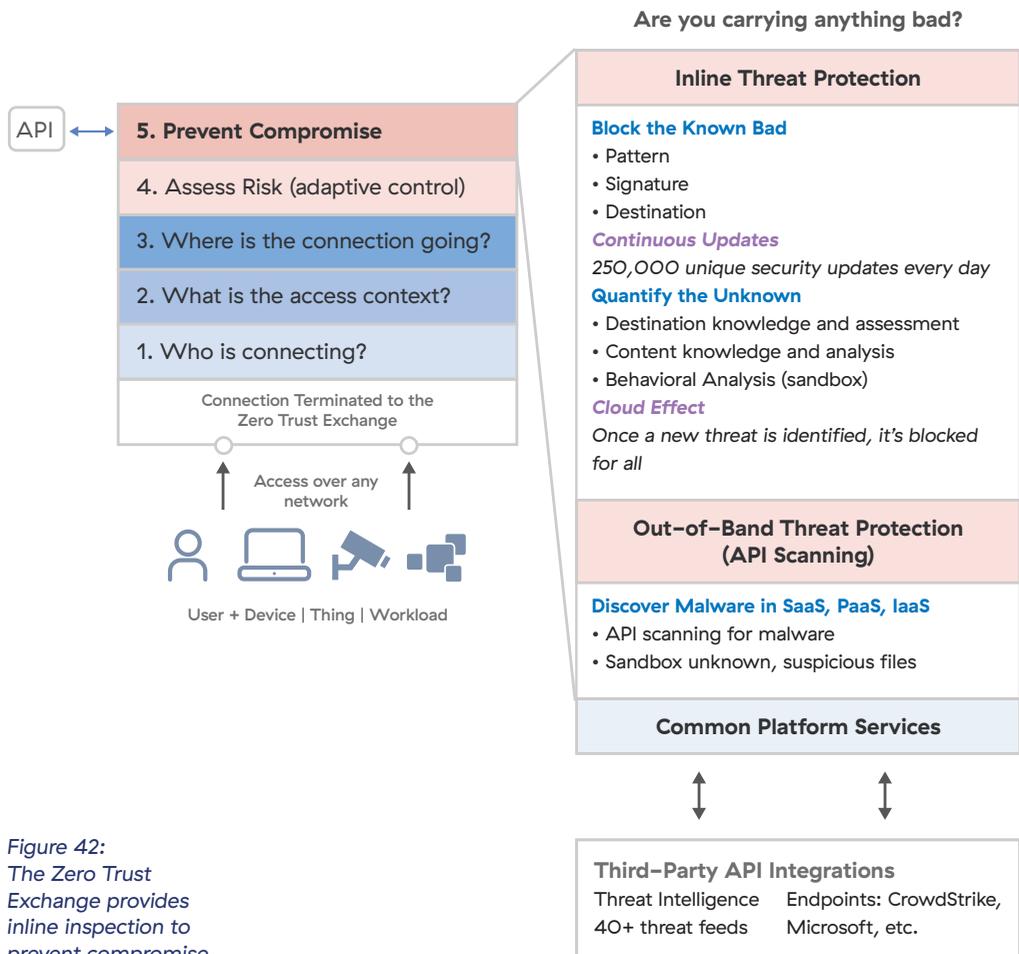


Figure 42:  
The Zero Trust Exchange provides inline inspection to prevent compromise.



Compromise prevention must take into account all threats targeting enterprises, which fall into two categories:

## Inline Threats

Malicious actors, code, plugins, and more also use SSL/TLS encryption as a means of transport. SSL/TLS public key encryption is the global standard for data protection of secure web transactions for the majority of the internet. Each data packet is turned into a code string decipherable only between an initiator and a destination service, regardless of the network.

This helps users protect sensitive information like passwords and credit card details, and prevents untrusted parties from observing or making sense of private communications. This protects against eavesdropping and data tampering by untrustworthy parties, but also gives threat actors the ability to hide their attacks.

To protect against threats via inline communication, enterprises must be able to do inline traffic decryption at scale.

## Out-of-Band Threats

It's important to also address the risks that are stored within SaaS, PaaS, IaaS, or other cloud solutions. An out-of-band assessment, as part of a unified threat management solution, provides enterprises with a full view of inbound threat paths and actively identifies threats before malware is downloaded, shared, or launched.



## Why is this important?

The ability to view traffic and cloud app use is critical for ensuring malicious content like botnets and malware isn't hidden in encrypted traffic. With the bulk of internet-bound traffic being encrypted, allowing this traffic to pass through unexamined or services to be used without inspection is risky. Inspection of both external and internal application access is critical since both traffic flows may be encrypted.

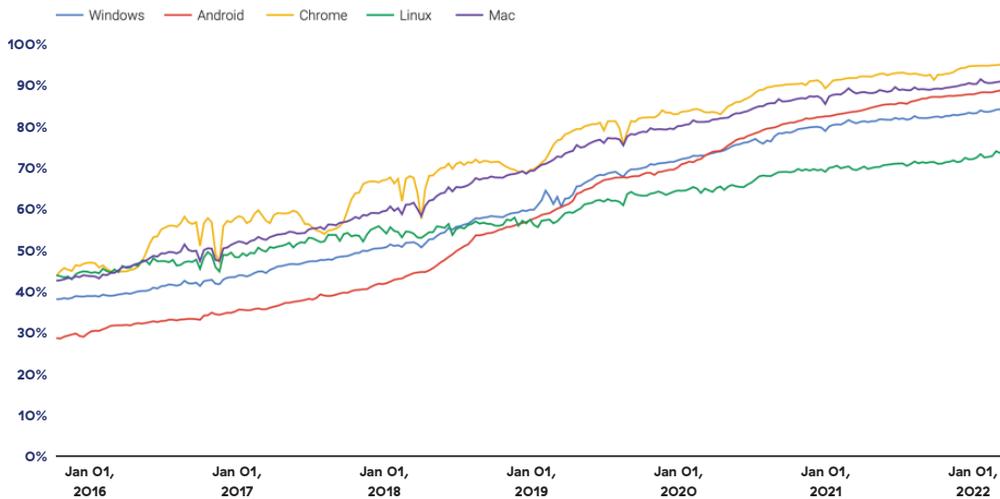


Figure 44: Growing rates of encryption used on the Chrome web browser.

Source: [Google Transparency Report](#)

Effective zero trust platforms employ comprehensive inline and out-of-band functions so customers can inspect content under proper circumstances, thus mitigating risks hidden in encrypted traffic or locales.

### Pro Tip:

Inspection in many geographies may take time and effort to find the right balance of privacy appropriate for workers' councils. Identifying the correct balance of risk reduction and privacy is not static and should be incremental, starting with less controversial geographies and traffic types.

Enterprises must consider the value of the visibility and insight garnered by inspecting traffic. This value must be assessed, however, in relation to privacy controls and restrictions for end users. Organizations must establish a balance between their right to be protected from threats and a user's right to privacy.

This balance must be considered and implemented granularly, not as a binary “inspect or don't inspect” policy. It should be implemented based on business risk and application type (see Element 2 for details on application categorization). Controls should provide protection where needed—e.g., to stop malicious files from being downloaded—while also ensuring end-user privacy is protected when personal data like healthcare or finance information is involved.

# Technology & Architecture Considerations

Visibility of enterprise threats requires the ability to uniformly look inside traffic in motion (inline) and files stored in services (out-of-band). Each discipline has different implementation considerations, but ideally should be managed with one set of policies.

## Inline considerations

The ability to inspect encrypted traffic requires a forward proxy architecture that's built for the cloud to allow for intensive inspection with minimal latency. For internet-bound traffic, decryption standards must support up to TLS 1.3 on all ports (which have improvements over earlier versions to ensure the confidentiality and integrity of communications) and check against techniques like DNS tunneling. Inspection also integrates with technologies like sandboxing, where potentially risky files can be “detonated” before being served to the user, as well as browser isolation, where pixels can be streamed to the user instead of the actual web page.

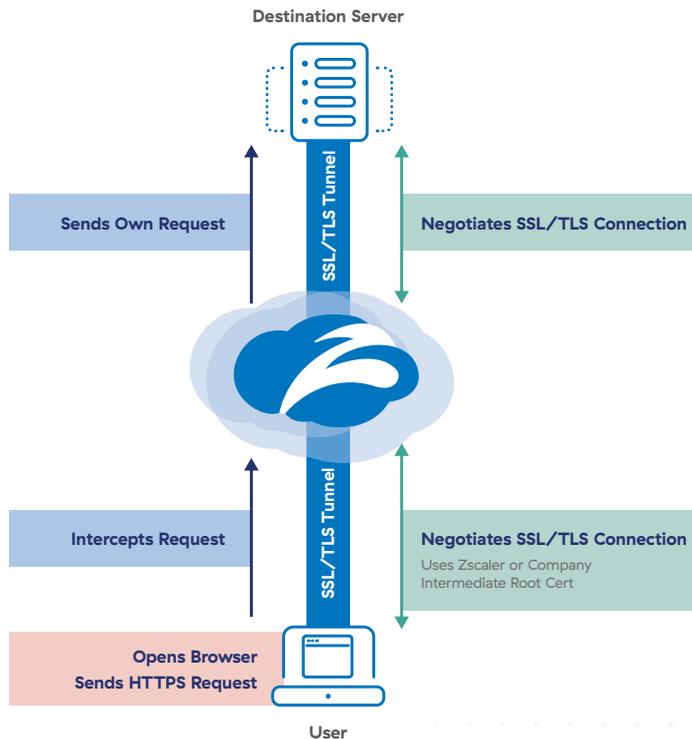


Figure 45: Process by which a client negotiates a secure session with a destination server.

Zero trust architecture must scale to function as an SSL/TLS, person-in-the-middle proxy that provides complete inbound and outbound content analysis able to immediately block threats detected anywhere in the enforcement plane. Threat actors continue to evolve their tools, techniques, and procedures, which include abuse of legitimate storage service providers like Dropbox, Box, OneDrive, and Google Drive for hosting malicious payloads.

In addition to stopping hackers, SSL/TLS inspection is useful when an enterprise wants to know when employees are intentionally or accidentally leaking organizational data. SSL/TLS inspection is also required for maintaining compliance by ensuring employees are not putting confidential data at risk (outlined in Element 6).

Therefore, true zero trust vendors must provide full inline SSL/TLS inspection capabilities. The most effective implementation of SSL/TLS inspection is through a native, proxy-based solution that is transparent to the end user. Bolting on a function to existing next-generation firewalls (NGFWs), which have inherent challenges scaling, is not recommended.

SSL/TLS decryption is resource-intensive. NGFWs with added inspection capabilities that have moved to virtual instances on CSP compute nodes will inevitably encounter limitations. Ensuring that data is secure at rest and not just in motion is also part of overall inspection controls (data at rest will be addressed in Element 6).

Implementing SSL/TLS inspection has been historically challenging for various reasons. Your own chosen zero trust vendor should be the foremost trusted expert and provide guidance, understanding, and implementation when enabling SSL/TLS inspection. Again, SSL/TLS inspection is non-negotiable for the SSE, but it need not sacrifice speed for security.

Method of SSL Inspection	Next-Gen Firewall	Proxy
How It Works	No termination of TCP connection. Stream-based (on the fly) scanning only.	Inline with two separate connections to client and server.
Impact of SSL Inspection	Can only see a small portion allowing malware to be delivered in segmented pieces. Needs additional proxy function (bolt on). Typically experience high performance loss when additional functionality (e.g., threat protection) is enabled.	Allows entire object to be reassembled and scanned. Allows for additional threat detection engines like sandbox and DLP.
TLS 1.3 Impact	Additional performance loss. Appliance refresh likely required to achieve claimed performance due to the higher performance and scale needs of new TLS 1.3 ciphers.	This is transparent and automatically handled by the Zero Trust Exchange.

Figure 46: A comparison of common SSL/TLS inspection techniques.

### Out-of-band considerations

Leveraging the same policy controls for inline inspection, a zero trust platform should govern data at rest within cloud applications, preventing dangerous file sharing and even file oversharing. When considered holistically, out-of-band controls complement previously outlined inline controls so cloud apps cannot be used as an attack vector.

Note: If the control assessment of Prevent Compromise cannot be met, the access must default, as outlined in Figure 11, to a Conditional Block policy.

## How does the Zero Trust Exchange accomplish this?

Zscaler is a true inline SSL/TLS proxy. It terminates SSL/TLS connections established by the client and establishes a new SSL/TLS connection to the server. From the client's perspective, Zscaler becomes the server and, from the original SSL/TLS server's perspective, Zscaler becomes the client. Since Zscaler is not only inspecting SSL/TLS traffic on the wire but also terminating the connections, Zscaler has full visibility to common name (CN) and other certificate parameters typically not visible to passive SSL inspection devices.

Zscaler's proven ability to inspect SSL/TLS has made it a Gartner-recognized industry leader for more than ten years. It was designed as a proxy architecture to enable low-latency throughput regardless of where an initiator or destination is located. The Zscaler Zero Trust Exchange inspection function is a cloud-native architecture that supports all current and future encryption requirements, including TLS 1.3 or earlier versions.

Zscaler's comprehensive set of inline protection with the Zero Trust Exchange provides a unique ability to understand the threats an enterprise faces. With these insights, Zscaler is able to offer a [global dashboard of threats](#), including all attacks Zscaler sees across its clouds.



Figure 47: Zscaler dashboards showing Cloud Activity and threats blocked.<sup>1</sup>

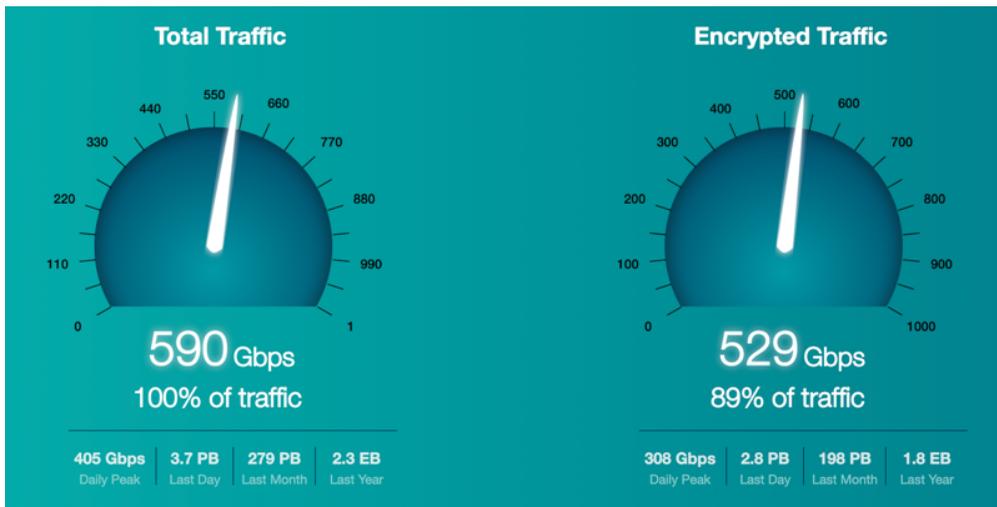


Figure 48: Zscaler dashboards showing traffic encrypted.<sup>2</sup>

Since data privacy is a common concern, Zscaler allows customers to granularly select which services to inspect. This allows Zscaler to protect against malware, for example, without delving into sensitive banking, healthcare data, etc., while delivering controls that are applicable to local laws and compliance requirements.

<sup>1</sup>Source: <https://www.zscaler.com/threatlabz/cloud-activity-dashboard>

<sup>2</sup>Source: <https://www.zscaler.com/threatlabz/encrypted-traffic-dashboard>

### Pro Tip:

Be sure to deploy inspection for SSL/TLS-encrypted data files. Too often inspections focus on the transport and not the content. This is especially important for common file formats like 7-ZIP, TAR, RAR, PDFs, and Microsoft Office files.

## What happens when you inspect SSL/TLS?

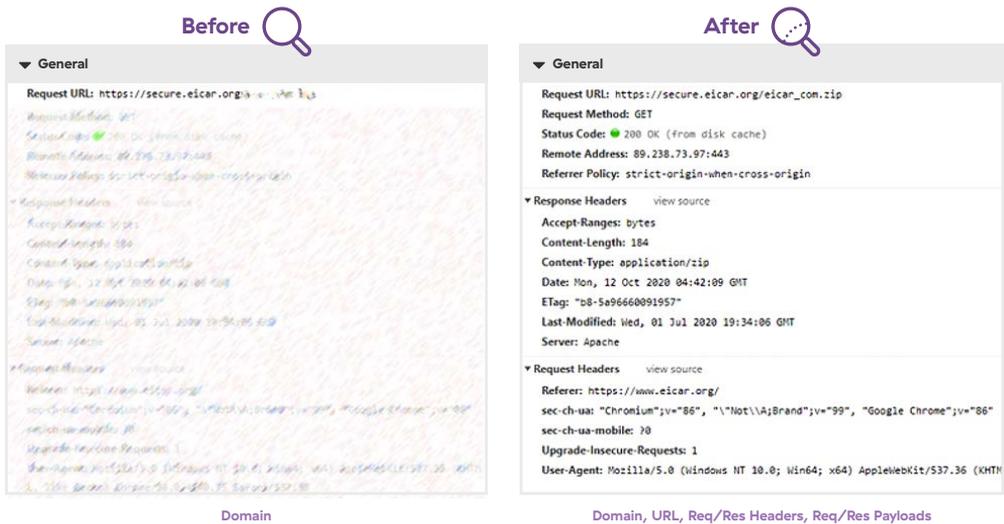


Figure 49: SSL/TLS inspection is critical to visibility for any security organization.

Traffic inspection has historically meant initiators and destinations sharing a network context, i.e., connected via the internet or a local network. Zscaler removes the need for a shared network context by connecting the initiator to the destination at the application layer rather than the network layer.

## The right control for the right application

The Zero Trust Exchange was built on the zero trust premise of ensuring only the correct controls are applied to the correct services. Ensuring these paths are accurate means effective connections and sessions for users, but more importantly, conserves resources. Enterprises must consider that not all traffic types can be, or should be, inspected. These must be addressed by each company under their specific use cases. Why send a video conference stream of real-time voice/video traffic via UDP (Zoom, Teams, etc.) through an inspection path, for example? There's nothing malicious within the video stream itself, so the goal should be to inspect the voice and video control plane traffic while bypassing real-time traffic without inspection.

Another common example is the use of pinned certificates. Often considered obsolete in terms of modern security designs, these are still employed in apps to ensure trust directly between the app and the client without relying on external certificate validation. If these apps are identified and want to be allowed, the appropriate bypasses should be set up to ensure that they function under the conditions for each enterprise.

Zscaler has deployed its inline inspection controls for two different sets of destination workloads:

### External applications

Internet destinations  
and SaaS

### Internal applications

Data center and  
IaaS/PaaS hosted

## External applications

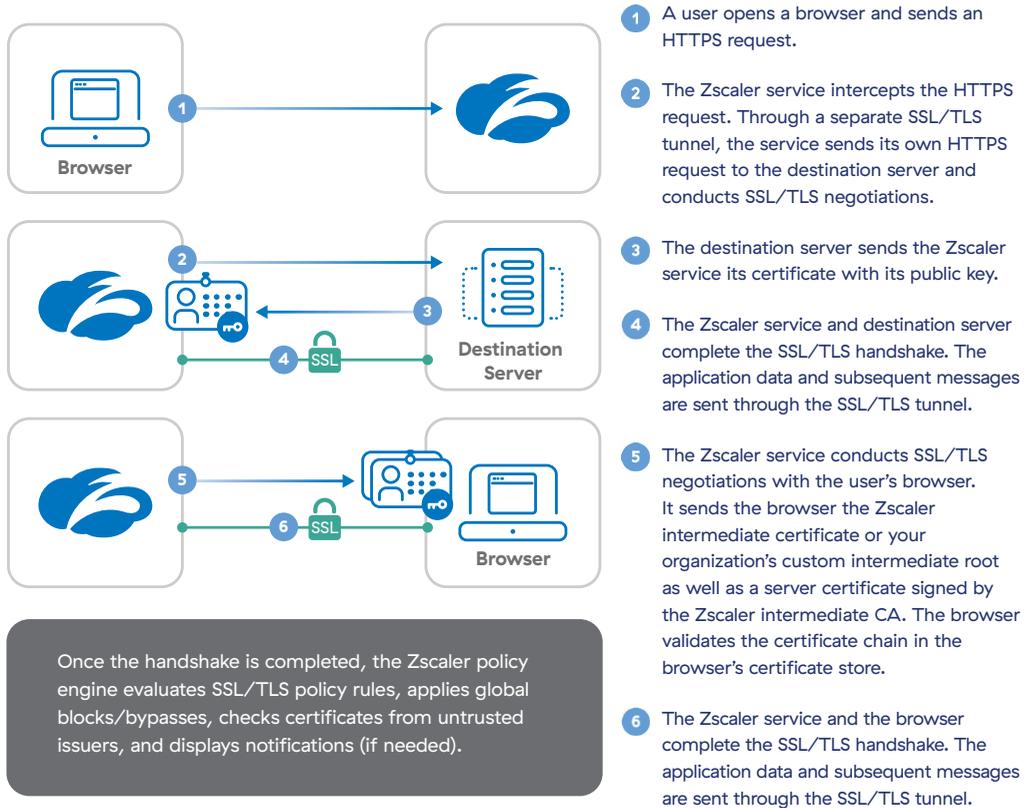


Figure 50: How Zscaler provides inline inspection for web applications.

The Zero Trust Exchange employs a single-scan, multiple access (SSMA) approach to inspection, allowing for multiple checks to be conducted on traffic that is entered into memory only once and providing significant performance enhancements.

Upon inspection of an internet-bound session with Zscaler, additional advanced threat protections can be simultaneously deployed to ensure the safety of an environment:

### **Secure Web Gateway (SWG)**

SWGs provide a safe, seamless web experience for enterprise end users. They eliminate ransomware, malware, and other advanced attacks in real time by leveraging the best of Zscaler's AI-powered analysis, URL filtering, and control.

### **Cloud Firewall**

Cloud firewalls extend protection to all ports and protocols for industry-leading protection by replacing edge and branch firewalls with a cloud-native platform.

### **Intrusion Prevention Systems (IPS)**

Fully inline, always-on IPS in the cloud allows any connection to receive inline control for stopping malicious activity from being delivered to the enterprise, including blocking activity like DNS tunneling.

### **Cloud Sandbox**

Advanced AI/ML-based cloud sandboxes stop patient-zero attacks with instant verdicts for common file types, automating the quarantine of high-risk, unknown threats.

Leveraging the entire set of post-SSL/TLS inspection tools allows enterprises to establish full visibility of all traffic both on and off the corporate network. These insights allow enterprises to protect against malicious attacks without performance impact or compromise. Insights aren't limited to user traffic, either.

The ability to inspect any internet-bound content has knock-on benefits, such as

- blocking unwanted, malicious server traffic so enterprise workloads can only access sanctioned services, protecting against incidents like the [SolarWinds compromise](#);
- identified IoT/OT services in the [ThreatLabz Internet of Things Dashboard](#) are updated live with content from Zscaler's entire cloud; and
- visibility and protection from threats including botnets, command-and-control, infiltration, etc., allow security operations greater context regarding affected infrastructure.

This inspection ability allows the Zero Trust Exchange to inspect the SSL/TLS traffic on the wire, but even as the Zero Trust Exchange terminates the connections, it retains full visibility over other certificate parameters not typically visible to passive SSL inspection devices. This makes the Zero Trust Exchange:

### Cloud scalable

Zscaler's custom TCP and SSL/TLS stack handles encrypted traffic on a global scale. Its architectural advantage ensures that SSL/TLS inspection becomes a non-issue for enterprises. Inspecting traffic, including encrypted traffic, in real time allows the Zero Trust Exchange to identify attacks while keeping an eye out for traffic that may include corporate secrets (see Element 4).

### Designed for the future

The majority of SSL/TLS traffic leverages TLS 1.2. While Zscaler has supported TLS 1.2 for years, it has also extended support for PFS with ECDHE and ECDSA ciphers. With Zscaler, supporting TLS 1.3 becomes a seamless change rather than a major configuration overhaul.

### Simple

Zscaler's "security-as-a-service" architecture operates seamlessly, without imposing new hardware planning requirements or costly upgrades to accommodate future TLS versions.

## Internal applications

When an initiator requests a service within the limits of your own security boundaries—VPCs or on-premises architecture, for instance—the threat model shifts. HTTP-based services must be inspected even after a valid user and acceptable risk score have been verified.

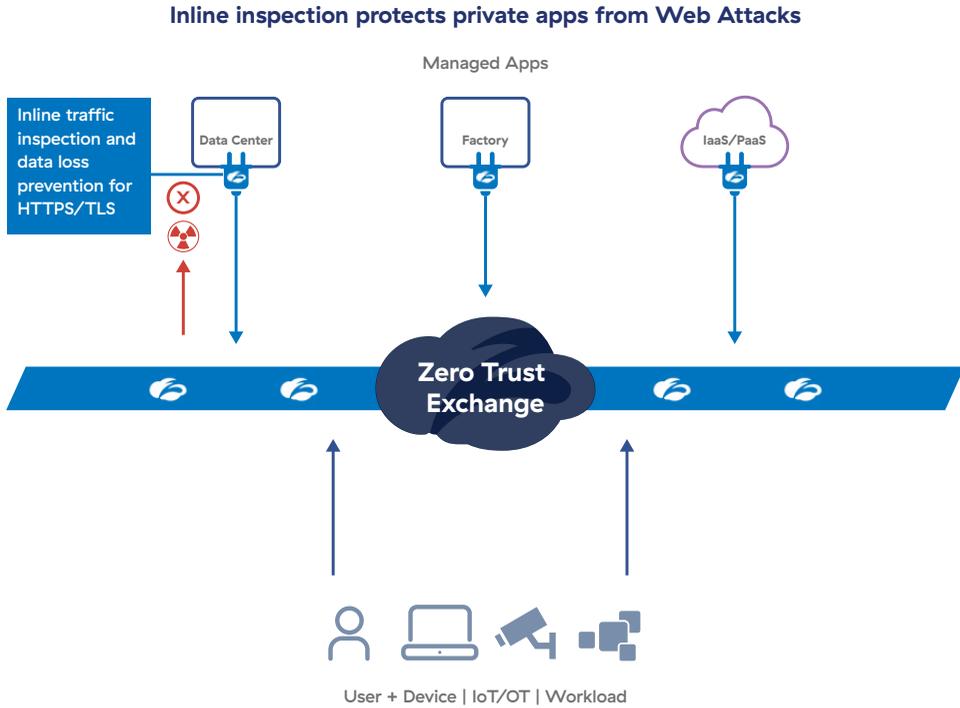


Figure 51: Protection of internal apps with inline inspection.

Zscaler is able to protect against the most prevalent web-borne attacks with inline inspection and prevention capabilities for these internal services. These capabilities run within each security boundary of an enterprise app, such that inspection for attacks on Segment A may differ from the inspection controls of Segment B, thus delivering granular control of functions based on the segment and access control needed.

Benefits of inspecting traffic passing through internal trusted services include

- shielding internal apps from web-based attacks stemming from malicious and non-compliant user traffic, such as [OWASP Top 10](#) attacks;
- detecting compromised or malicious user accounts, since authorized users or devices should not be attacking internal services; and
- the ability to build customized rules to protect enterprise-specific environments.

The Zero Trust Exchange is a complete data protection platform built for both inline and API-based (out-of-band) inspection. It provides detailed visibility of data at rest and in motion to help teams make better data protection decisions and quickly identify malicious content that may be stored in cloud repositories. Further discussion of data protection is provided in Element 6.

Inline protection acts as the building block for establishing which data should travel versus which shouldn't. Out-of-bound controls allow enterprises to apply controls to counter any threats against data at rest.

# Zero Trust Progress Report

After the SSL/TLS inspection and threat prevention phase, the zero trust process has oversight of the applications the initiator has accessed. The contents of this view allow further insights to be applied to the policy decision.



Jane Doe



Application X

**Prevent Data Compromise**  
Ransomware and unwanted download blocked

**Compute Dynamic Risk Score**  
**User Posture** – Untrusted network  
**Device Posture** – AV only  
**Behavior** – Repeated login attempts  
*Risk Score – High*

**Verify where the access is going**  
 Internal ERP service on port 23 in clear text  
 External new Social Media site on port 443, encrypted

**Verify Access Conditions**  
**What is the user profile:**  
 Untrusted location, using personal device,  
 browser access

**Verify Identity**  
**User Identity and Device:** Jane Doe, 2FA, consultant



**Jane Doe wants to access Application X**



John Doe



Application Y

**Prevent Data Compromise**  
Nothing detected

**Compute Dynamic Risk Score**  
**User Posture** – Trusted network  
**Device Posture** – AV/EDR, Disk Encrypted  
**Behavior** – Normal  
*Risk Score – Low*

**Verify where the access is going**  
 Internal File share service on port 445  
 External video sharing site on port 443, encrypted

**Verify Access Conditions**  
**What is the user profile:**  
 Private location, using enterprise device,  
 Client Connector

**Verify Identity**  
**User Identity and Device:** John Doe, MFA, internal



**John Doe wants to access Application Y**

Progress Report 5: Inspection of traffic reveals and blocks malicious content for Jane.

# Element 6: Prevent Data Loss

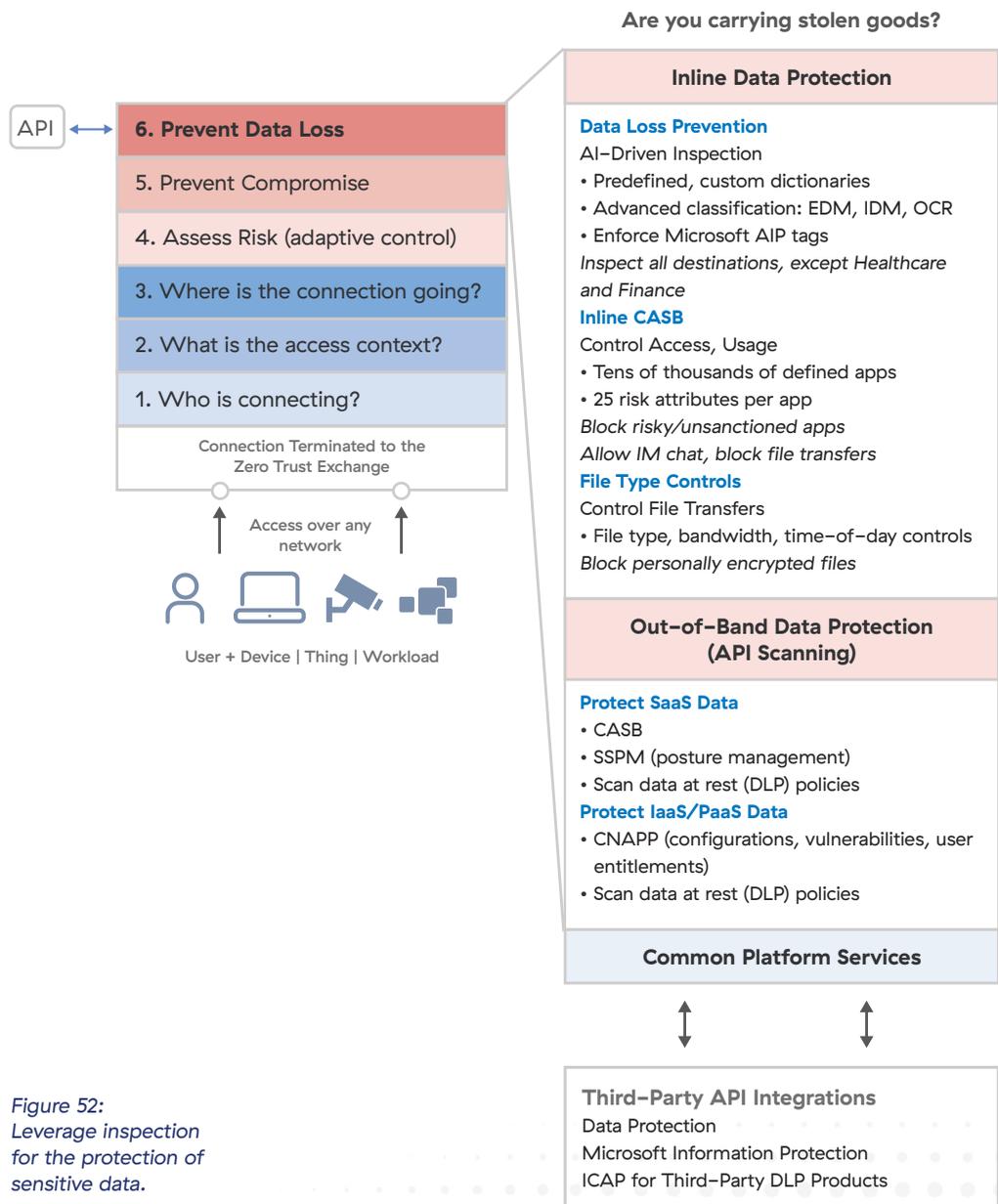


Figure 52:  
Leverage inspection  
for the protection of  
sensitive data.

As part of the Control phase, enterprises must consider what data leaves the organization. SSL/TLS inspection at scale ensures that attacks against an enterprise are stopped, and this same inspection can also be applied for data egress controls.



### Why is this important?

The ability to inspect traffic destined for the internet, SaaS, or internal applications is important for identifying and preventing the loss of sensitive data. The use cases are obvious for the internet, where both inline decryption and inspection and out-of-band API scanning must ensure that sensitive data is not leaked or exfiltrated to unauthorized cloud services. However, the same protections should be extended to internal application access. This capability applies to both managed and unmanaged devices and is also important when considering IoT/OT devices and workload communications.

As outlined in the previous element, over 80% percent of web traffic is encrypted, including common tools used by both enterprise and private users such as the file-sharing services Dropbox and Google Drive, or collaboration tools like Microsoft 365 and Google Chat. If this traffic is completely encrypted, enterprises are powerless to understand what's being sent to these environments. This means businesses are unable to protect against both inbound malicious payloads and illicitly outbound intellectual property.

However, concerns are not limited solely to users. As the [2020 SolarWinds breach](#) showed, enterprises not monitoring server data sent outbound to the open internet are unlikely to be able to stop the exfiltration of sensitive data, e.g., via a supply chain attack.

It is thus critical that once SSL/TLS interception is enabled, an enterprise deploys the necessary inline controls to protect against information and intellectual property leakage. Traffic inspection allows enterprises to identify not only what intellectual property is being accidentally shared but also enables greater identification and protection against ransomware attacks.

Zscaler’s [2022 Ransomware Report](#) documented a substantial increase in attackers now exfiltrating data as part of double extortion attacks. Attackers are now asking for ransoms not only to decrypt enterprise systems but are also increasingly stealing intellectual property only to ransom it back to the same company. Figure 53 gives a high-level view of the rate of change from 2020 to 2021.

### Percent change in double extortion attacks: 2021 vs. 2020

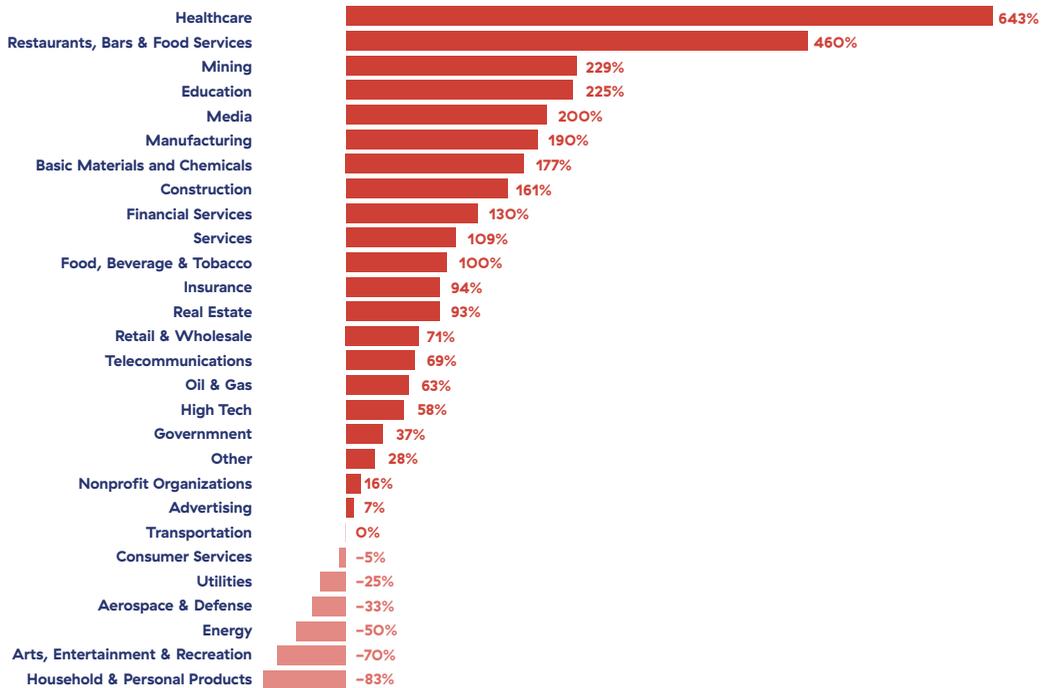


Figure 53: Industry breakdown of double extortion attacks in 2021 compared to 2020 from Zscaler’s Ransomware Report.

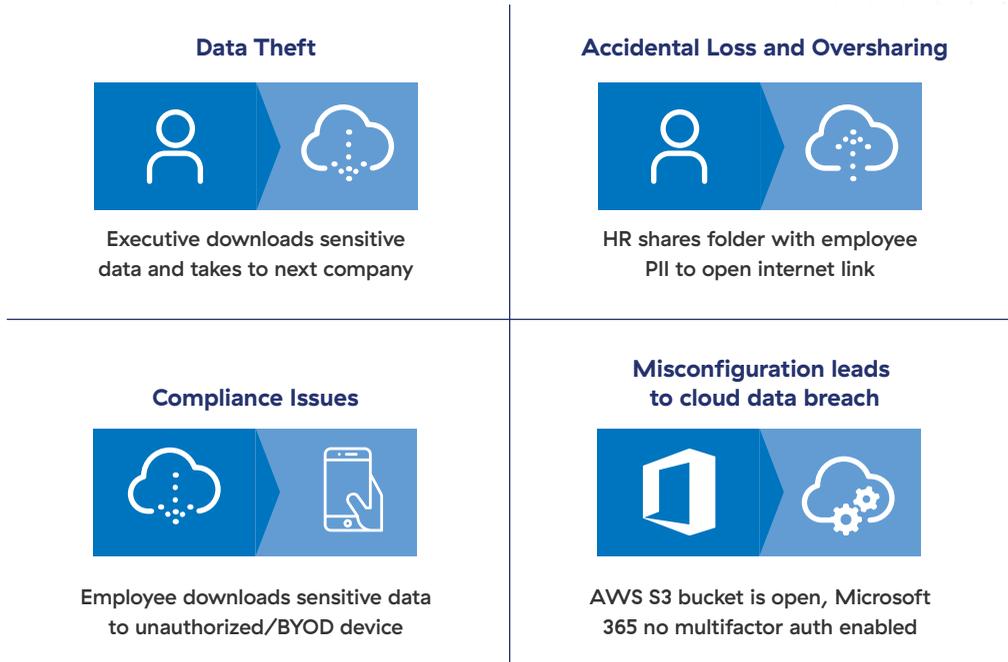


Figure 54: Business challenges due to lack of data protection.

Again we arrive at the problem of determining the correct balance between inspection and privacy. This review must be understood in terms of enterprise risk.

## Technology & Architecture Considerations

Widespread adoption of cloud applications means organizations' sensitive data is widely distributed. The top two enterprise data exfiltration channels are cloud desktop and personal email applications. Adequate protection technologies should deliver complete contextual visibility and enforcement capabilities when rogue users upload sensitive data to personal cloud desktops. They should also stop data exfiltration on personal and unsanctioned webmail services like Gmail and Outlook.

Protection starts with blocking access to unauthorized users. This is the simplest protection policy. Consider two examples:

- **For users and devices** – Implementing controls against destinations that contradict corporate policy, such as webmail services.
- **For IoT/OT devices and workloads** – Restricting workload or OT services to communications with relevant services such as patching solutions, or preventing a workload's access to unnecessary services such as YouTube.

The protection outlined in Element 3 must be able to seamlessly create sets of common apps that relate to a singular service—for example, Microsoft 365 is a set of separate applications that can be addressed as a group or individually. These protection solutions must also be able to differentiate between personal and professional applications and cloud services and apply sets of rule definitions appropriately. As an example, a CEO should be able to upload financial reports to a corporate SharePoint file store, but unable to upload the same files to a personal file sharing service.

**Inspect all content with full inline SSL inspection**  
On/Off Network — All Ports/Protocols

		
<p><b>Context and Content-Aware DLP</b></p> <ul style="list-style-type: none"><li>• Predefined dictionaries and engines for PCI, PII, PHI, GDPR, etc.</li><li>• Custom dictionaries via keywords and patterns</li></ul>	<p><b>Advanced Data Classification</b></p> <ul style="list-style-type: none"><li>• Exact Data Match: Match on custom data to reduce false positives</li><li>• Index Document Match: Fingerprint sensitive documents for blocking</li><li>• Optical Character Recognition: Find text in images and block</li></ul>	<p><b>Azure Information Protection (AIP) Integrations</b></p> <ul style="list-style-type: none"><li>• Enforce AIP tagging for data inline and inside SaaS apps</li><li>• Write AIP tags to sensitive data found inside SaaS apps</li></ul>

Figure 55: Functions required for an effective data protection solution.

Enterprise resource protection must also include the ability to view resources that may be out-of-band, which entails scanning the APIs of SaaS providers to protect data at rest, or inline, which requires the scanning of data in motion.

**Pro Tip:**

Building data protection libraries can be intimidating. Start with predefined dictionaries, but also ask what types of data, like intellectual property, shouldn't be exposed to competitors.

Inspection and subsequent protection policy application must happen at scale, wherever the user is located. It cannot be centralized in a data center or destination cloud if it's to pass through at scale.

Upon decryption, protection policies must deliver appropriate controls based on the identity and app access permissions verified in Elements 1, 2, and 4. This protection policy includes two key areas, data in motion and data at rest.

### **Data in motion (inline controls)**

Information passing from the initiator to a destination application must be reviewed to ensure it does not contain sensitive corporate information. With inline inspection, enterprises can prevent data from being uploaded to unsanctioned apps, being downloaded to unauthorized devices, and malicious content from being downloaded or uploaded.

### **Data at rest (out-of-band controls, independent of SSL/TLS inspection)**

Data stored in cloud services must also be assessed to mitigate corporate information leakage. This should be queried via APIs to identify. Cloud access security broker (CASB) solutions offer these out-of-band controls for granular access based on a rich set of cloud app definitions, file type controls, and risk attributes.

**Note:** Any file store that leverages file-level encryption like AES would be visible to an out-of-band API as an encrypted file only. There is no visibility within the contents of the file.

Both of these functions must rely on protection solutions including:

- **File type control** – Define acceptable files for enterprise access
- **Cloud app control** – Determine which cloud apps are allowed and which are not
- **Machine learning-enabled DLP** – Improve the accuracy of sensitive data departure detection
- **Exact data match** – Index structured data (for example, your specific credit cards versus a 16-digit number that algorithmically could be a credit card, etc.) to help reduce false positives
- **Optical character recognition** – Block sensitive data found in images or screenshots
- **Indexed document matching** – Protect high-value documents containing sensitive information like intellectual property

These protection solutions empower enterprises to break an attacker's kill chain, as outlined in the white paper [Transforming Cybersecurity Response with Zscaler using the MITRE ATT&CK Framework](#).

Note: If the control assessment of Prevent Data Loss cannot be met, the access should default, as outlined in Figure 11, to a Conditional Block policy.

## How does the Zero Trust Exchange accomplish this?

Protection is more than simply cloud access security. The Zero Trust Exchange follows users wherever they go, ensuring that data protection controls are applied even when users connect direct-to-cloud. This provides a range of benefits that can only be delivered by a global, purpose-built security cloud.

The Zero Trust Exchange is a complete data protection platform built for both inline and API (out-of-band) inspection. It provides granular visibility of data at rest and in motion to help teams make better data protection decisions and quickly identify malicious content that may be stored in cloud repositories or unauthorized uses of shadow IT. Essentially, inline data protection acts as the building block for establishing the paths data should travel versus the ones it shouldn't.

With the Zero Trust Exchange in place to control what data should leave your network and what sanctioned apps need to be secured, enterprises can start considering out-of-band CASB for protecting data at rest. This prevents sensitive information from being shared via open internet links or handed over to unauthorized groups. Out-of-band controls can scan cloud apps for dangerous malware and leverage AI/ML-enabled sandboxing to quickly identify files that shouldn't be mixed in with sensitive data.

Inspection and prevention policy application happens at scale regardless of user location through a single, globally distributed platform without the need for multiple prevention solutions to be cobbled together. This gives enterprises full visibility and control of any and all intellectual property or sensitive content being in motion or at rest. These Zero Trust Exchange functions are built on integrations across various SaaS applications and public clouds.

Zscaler implements technologies to identify such assets, including:

**Visibility of and insight into cloud apps**

- Tens of thousands of cloud app definitions
- Multiple levels of risk attributes per app definition

**Data loss prevention**

- File type/size control
- Predefined dictionaries and engines
- AI/ML dictionaries and engines
- Custom dictionaries
- Reg-ex and proximity filters
- Advanced classification
- Exact data match
- Indexed document matching
- Optical character recognition
- AIP/MIP integration

**User and Entity Behavior Analytics (UEBA) (which also contribute to risk scores discussed in Element 5)**

- AI- and ML-based analytics
- Threshold-based user anomalies

**SaaS and Cloud Security Posture Management (SSPM/CSPM)**

- Evaluate SaaS and IaaS configurations and permissions to automatically remediate issues

**Out-of-band (API-driven) CASB**

- Predefined, customizable DLP dictionaries for SaaS and public clouds like AWS
- Collaboration management searches for risky file shares and revokes access
- Cloud sandbox scans data at rest to identify and respond to zero-day malware and ransomware
- Zscaler's Browser Isolation streams pixels instead of actual web content to protect data

### Secure User Access to Data

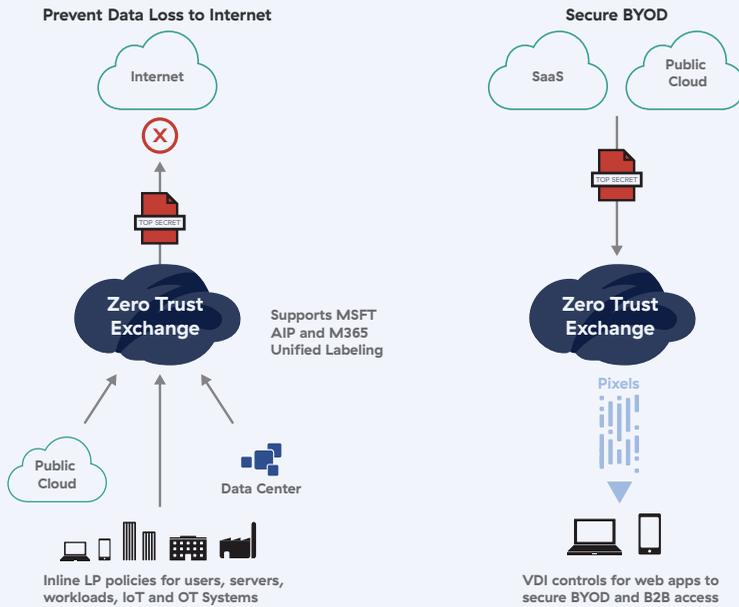


Figure 56: The Zero Trust Exchange provides secure user access to data.

### Secure Data in Public Cloud and SaaS

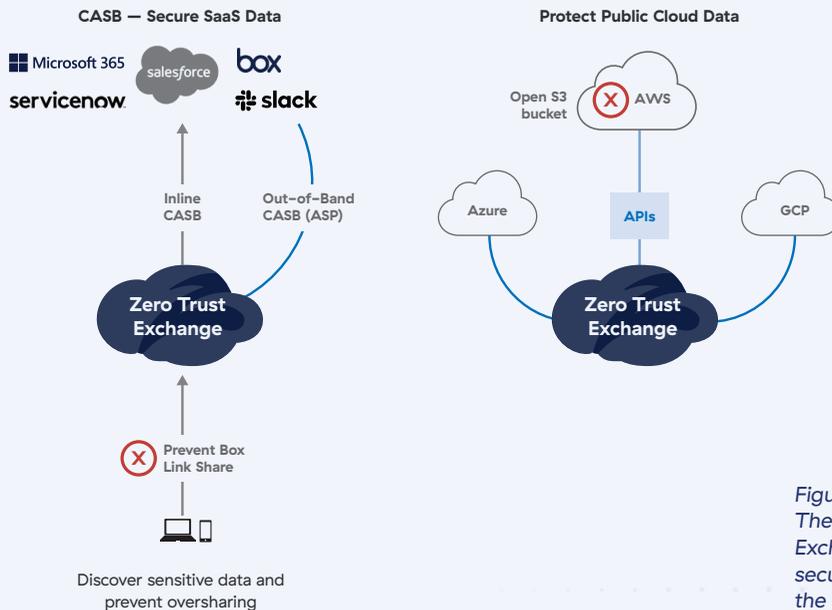


Figure 57: The Zero Trust Exchange also secures data in the public cloud.

Zscaler can also provide secure, controlled access for unmanaged devices (BYOD) requesting information. But what happens to the information once it's downloaded to an unmanaged device? Organizations must control how this data is accessed and where it resides. Thus, Zscaler's Browser Isolation enables access to private data without allowing the data to persist on the device by streaming pixels rather than allowing unfettered browser access.

Beyond inspection of traffic and content, Browser Isolation can deliver a safe gap between users and the internet and SaaS-based web apps that they connect to. By rendering app content as a stream of images, Browser Isolation allows enterprises to deliver access to services to valid users without ever allowing direct connections to the app itself. This eliminates the possibility of data leakage or threat delivery. Zscaler delivers isolation for both external and internal services as outlined in Element 7 covering policy enforcement.

This comprehensive set of inline protection controls gives the Zero Trust Exchange the unique ability to provide views of who is accessing what on the internet. Given this capability, Zscaler is able to offer reports including:

- [A Shadow IT report](#): A per-tenant view of unknown, internet-connected devices
- [An IoT Dashboard](#): A global view of IoT services using Zscaler's cloud
- [Cloud Application Dashboard](#): Insight into all cloud apps consumed across Zscaler's clouds

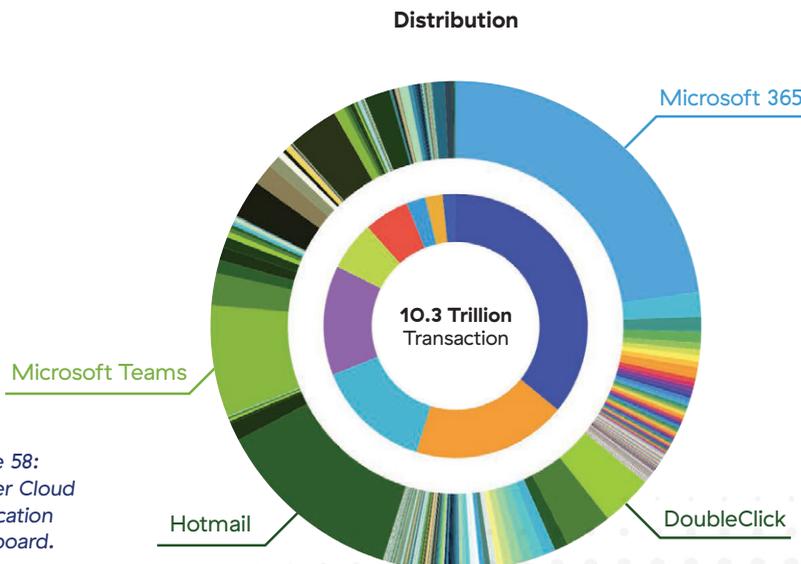


Figure 58:  
Zscaler Cloud  
Application  
Dashboard.

# Zero Trust Progress Report

In assessing what information should be protected for each session, the control section identifies anything that may be unnecessarily exposed, then exercises the appropriate controls.



Jane Doe



Application X

<p><b>Prevent Data Loss</b> Nothing detected</p>
<p><b>Prevent Data Compromise</b> Ransomware and unwanted download blocked</p>
<p><b>Compute Dynamic Risk Score</b>  <b>User Posture</b> – Untrusted network  <b>Device Posture</b> – AV only  <b>Behavior</b> – Repeated login attempts  <i>Risk Score – High</i></p>
<p><b>Verify where the access is going</b>                      Internal ERP service on port 23 in clear text                      External new Social Media site on port 443, encrypted</p>
<p><b>Verify Access Conditions</b>  <b>What is the user profile:</b>                      Untrusted location, using personal device,                      browser access</p>
<p><b>Verify Identity</b>                      User Identity and Device: Jane Doe, 2FA, consultant</p>



**Jane Doe wants to access Application X**



John Doe



Application Y

<p><b>Prevent Data Loss</b> Nothing detected</p>
<p><b>Prevent Data Compromise</b> Nothing detected</p>
<p><b>Compute Dynamic Risk Score</b>  <b>User Posture</b> – Trusted network  <b>Device Posture</b> – AV/EDR, Disk Encrypted  <b>Behavior</b> – Normal  <i>Risk Score – Low</i></p>
<p><b>Verify where the access is going</b>                      Internal File share service on port 445                      External video sharing site on port 443, encrypted</p>
<p><b>Verify Access Conditions</b>  <b>What is the user profile:</b>                      Private location, using enterprise device,                      Client Connector</p>
<p><b>Verify Identity</b>                      User Identity and Device: John Doe, MFA, internal</p>



**John Doe wants to access Application Y**

Progress Report 6: Evaluation of data loss.

**Section**

**3**

**Enforce**

## Enforce

Policy, Per-Session  
Decision and Enforcement

### 7. Enforce Policy

The Verify and Control stages present a status snapshot in time, but identity and context must be constantly evaluated before policy is applied. The policy engine considers the output of a dynamic assessment, which includes all necessary categories, criteria, and insights pertaining to each access request.

Policy can only be enforced precisely when the full context of the user, app, device posture, etc., informs the decision. The policy outcome is to either allow or restrict access based on this context.

# Element 7: Enforce Policy

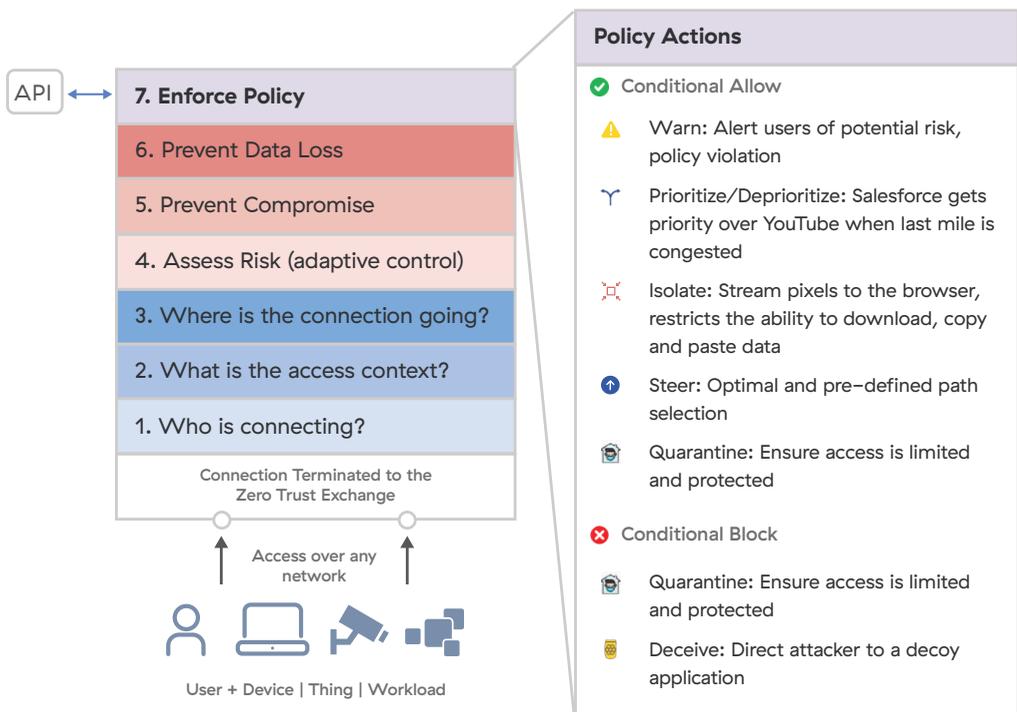


Figure 59:  
Access policy is enforced based on a number of Conditional Allow or Conditional Block actions.

Following the Verify and Control stages, and armed with an understanding of dynamic risk assessments, we arrive at the point of enforcement. Enforcement is not centralized on a network of dedicated equipment, as is often the case with traditional security solutions. Authorization decisions made in Element 1 and the assessments in Element 2 influence enforcement in our current stage.



### Why is this important?

Policy enforcement must be constantly and uniformly applied. This is only possible when policy remains the same, and is applied equally regardless of location of the enforcement point.

# Technology & Architecture Considerations

Zero trust allows us to start at zero. Access is denied to anyone or anything as an initial condition. This means no services, FQDNs, or IPs are exposed or can be discovered. An initiator can only connect to their requested destination as befitting their access criteria.

This type of invisibility–until–permission is only possible if policy is enforced independent of the network and, most importantly, if the initiator and the destination are abstracted from one another.

Policy enforcement must take into consideration all previously validated qualities:

1. Identity
2. Context
3. Application destination
4. Dynamic risk
5. Malicious content
6. Data loss

## Policy Enforcement Actions

Following any decision, enforcement takes on one of the following actions, either conditional allow or conditional block.

### Conditional Allow

Access is granted to the requested application not as a blind allow–and–connect, but with the ability to deliver additional controls, such as isolation, inspection, and warnings.

### Conditional Block

If conditions of an access request are flagged in Elements 1–6, access is blocked. Block controls can be called at any time during Elements 1–6 if an initiator fails one of the tests. For example, access would be blocked if an authorized user and device are validated, but then malware is downloaded.

## How does the Zero Trust Exchange accomplish this?

Traditional, network-based solutions connect initiators to a shared network, either physically on a local network or logically via a virtual private network (VPN). This extends the network to wherever a user is connected, and control is applied at the network layer.

Anyone on that network can see all other nodes. A control, such as a firewall, can minimize some of the attack surface, but if access is needed to a service, there still needs to be an open path through the controls. This is similar to Figure 60 below where standing on a street allows an onlooker to see each house, approach it, and test if their key works in the door.



Figure 60: Traditional network controls preserve visibility so anyone can see all of the houses and doors.

The Zero Trust Exchange enforcement policy allows for various controls to be enforced based on the following formula:



Figure 61: The valid inputs from Elements 1–6 allow for accurate and granular policy to be enforced.

Enforcement within the Zero Trust Exchange is not simply limited to the two options of “conditional allow” or “conditional block.” Its controls are at the application layer, not at the network layer as with legacy network control solutions. The Zero Trust Exchange is implemented as a proxy for all applications and thus allows for policy control at a very granular level.

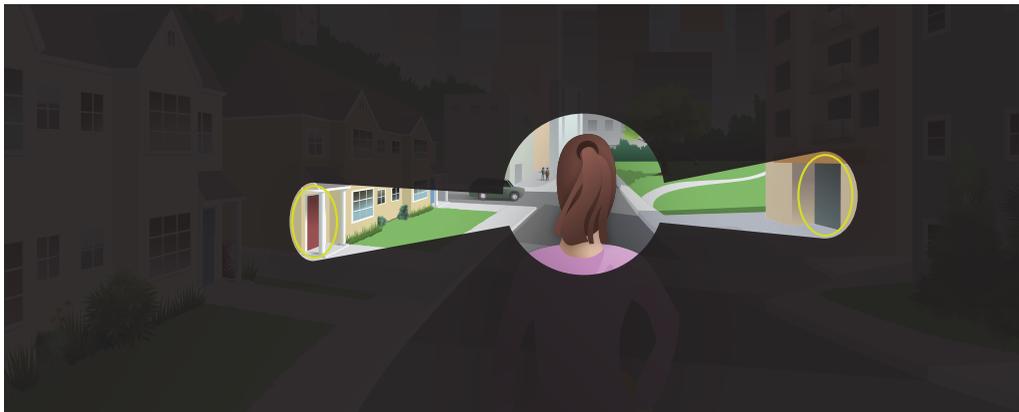


Figure 62: Ensuring users can access what they need and nothing more is key to zero trust.

The Zero Trust Exchange provides numerous Conditional Allow and Conditional Block outcomes:

## Conditional Allow

**Allow:** If all elements are answered, then the Zero Trust Exchange will allow traffic to pass.

**Warn and Allow:** Similar to the Allow outcome, a warning allows a company to state that, while access is allowed, the risk of the initiator or the destination is not clear. The enterprise can then prompt the initiator to be aware of the risk and then continue.

**Prioritize (bandwidth control):** Empowers enterprises with dedicated network links to preserve access to business-critical applications regardless of network bandwidth consumption. This allows the creation of rules to prioritize business-critical application traffic.

**Deprioritize (bandwidth control):** Empowers enterprises with dedicated network links to preserve access to business-critical applications regardless of the internet pipe consumption. This allows for the creation of restrictive rules around social media and streaming media.

**Isolate:** This creates a safe gap between users and the web, rendering content as a stream of pixels to eliminate data leakage and the delivery of active threats.

**Steer:** This decision instructs on how to send traffic. Steering is a mechanism for sending traffic to non-standard destinations. This could be for leveraging an anchored IP address for geo-specific solutions or sending traffic through a more effective path.

**Quarantine and Allow:** This result uses cloud sandbox and AI/ML to identify potentially harmful content, which is then “detonated” in a safe environment. If benign, the connection is granted.

## Conditional Block

**Block:** If the conditions of your access requests do not pass the evaluations of Elements 1–6, then the Zero Trust Exchange will block the session.

**Deceive:** Deception is a proactive defense approach that detects active threats by populating an environment with decoys: fake endpoints, files, services, databases, users, computers, and other resources that mimic production assets for the sole purpose of alerting of an adversary's presence.

**Quarantine and Block:** This uses cloud sandbox and AI/ML to identify potentially harmful content, which is then “detonated” in a safe environment. If dangerous, the connection is blocked.

The Zscaler Zero Trust Exchange allows for multiple enforcement controls to be applied in policy and not simply make a binary decision of allow or block. For example, if users are connecting to an external portal that has a source IP trust requirement, a Zscaler policy could be built to (1) steer traffic from an egress IP address, (2) prioritize traffic, and (3) isolate the session in a web browser.

This multilayered policy enforcement delivers powerful controls for enterprise protection and decisions. Enterprises can build various levels of policy enforcement based on the outcomes of the previous six elements. These policies should enforce business outcomes, risk mitigation, and security posture, as examples.

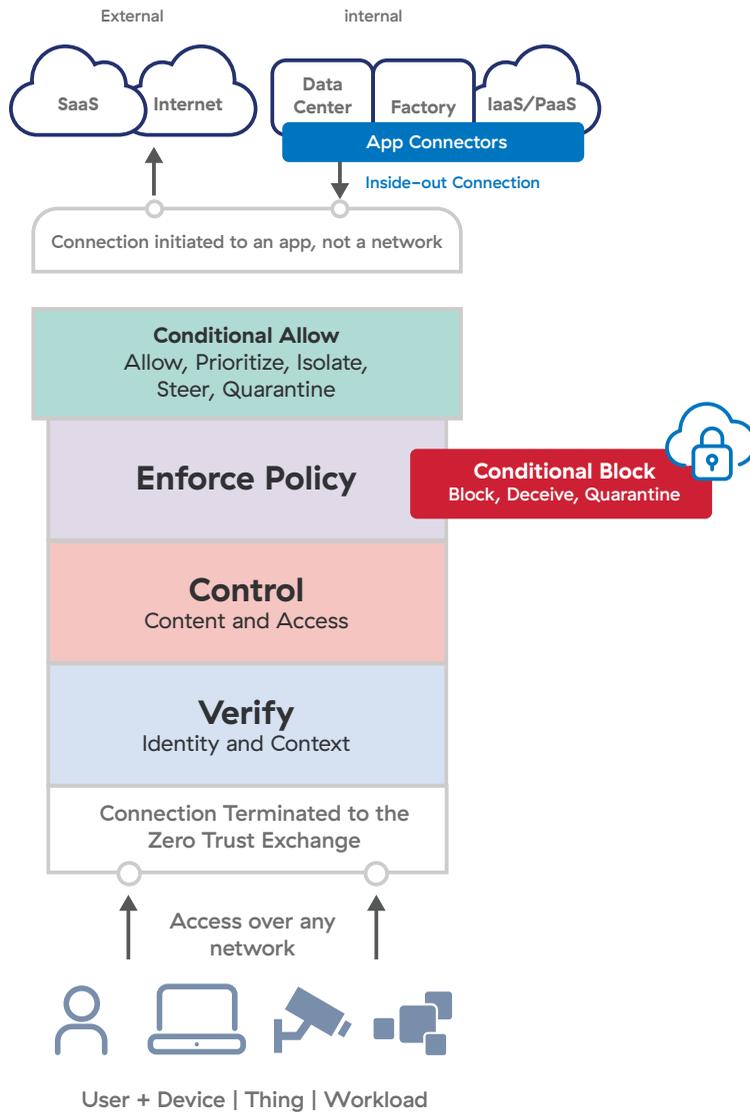


Figure 63:  
Policy enforcement either  
“conditional allow” or  
“conditional block.”

# Zero Trust Progress Report

Upon the completion of the policy enforcement phase of the Zscaler Zero Trust Exchange:

- All elements will have been evaluated
- Assessments are completed
- Scores are assigned
- Access has been conditionally allowed or conditionally blocked

Should access be granted, then the last step for access is the ability to get traffic to the correct destination application.

 Jane Doe
 Application X

**ENFORCE ACCESS POLICY: BLOCK**

<b>Prevent Data Loss</b> Nothing detected
<b>Prevent Data Compromise</b> Ransomware and unwanted download blocked
<b>Compute Dynamic Risk Score</b> <b>User Posture</b> – Untrusted network <b>Device Posture</b> – AV only <b>Behavior</b> – Repeated login attempts <b>Risk Score</b> – High
<b>Verify where the access is going</b> Internal ERP service on port 23 in clear text External new Social Media site on port 443, encrypted
<b>Verify Access Conditions</b> <b>What is the user profile:</b> Untrusted location, using personal device, browser access
<b>Verify Identity</b> User Identity and Device: Jane Doe, 2FA, consultant



**Jane Doe wants to access Application X**

 John Doe
 Application Y

**ENFORCE ACCESS POLICY: ALLOW**

<b>Prevent Data Loss</b> Nothing detected
<b>Prevent Data Compromise</b> Nothing detected
<b>Compute Dynamic Risk Score</b> <b>User Posture</b> – Trusted network <b>Device Posture</b> – AV/EDR, Disk Encrypted <b>Behavior</b> – Normal <b>Risk Score</b> – Low
<b>Verify where the access is going</b> Internal File share service on port 445 External video sharing site on port 443, encrypted
<b>Verify Access Conditions</b> <b>What is the user profile:</b> Private location, using enterprise device, Client Connector
<b>Verify Identity</b> User Identity and Device: John Doe, MFA, internal



**John Doe wants to access Application Y**

Progress Report 7: Jane's request is blocked based on an evaluation of the 7 elements, while John's access is allowed.

# Connecting to the Applications

## Connecting to the Applications

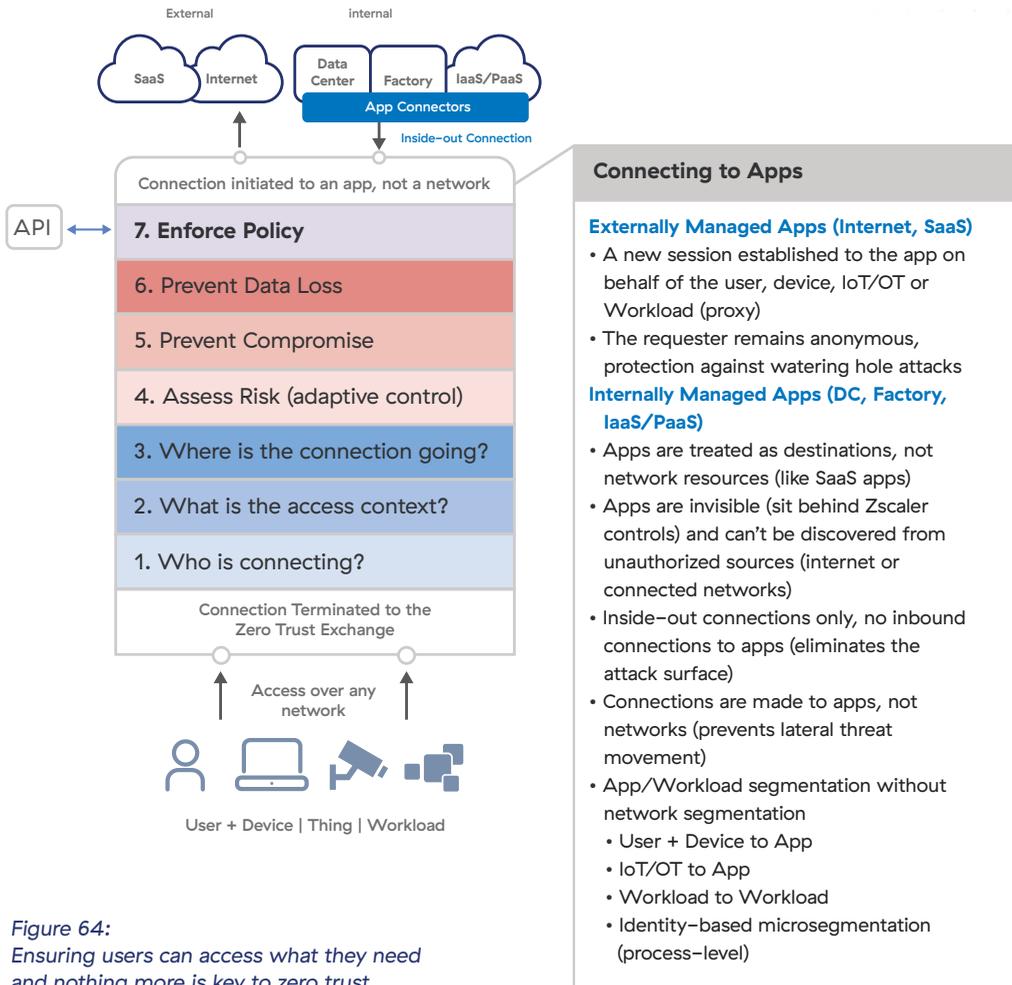


Figure 64:  
Ensuring users can access what they need and nothing more is key to zero trust.



## Why is this important?

Remember, policy enforcement must be constantly and uniformly applied, and this is only possible when policy is applied equally no matter where the initiator is based or where the applications are hosted.

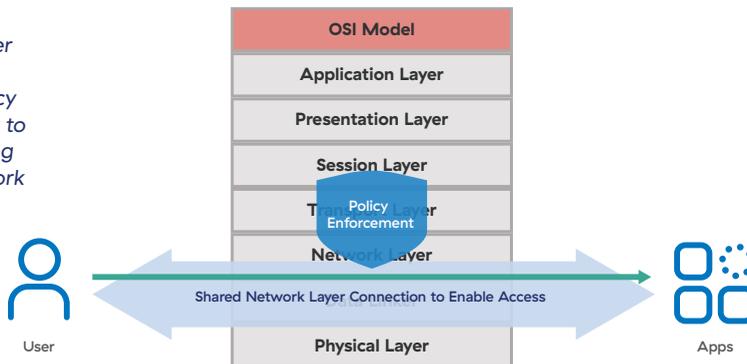
## Connecting to the Applications

Take your house or apartment, for example. The locks on your doors are for your doors only. You can't use your keys to enter someone else's house and, conversely, people can't use their keys on your locks. Each key is meant to be unique. This idea of local control has long been utilized within traditional security controls.

The network to which a user was connected historically determined security control. The challenges for this sort of control become obvious when a user moves from one network to another. Once the "other network" is being used, controls can no longer be applied to the user. Again, it's like taking your own keys and trying to use them to open doors on another house.

It is highly inefficient to move network controls and infrastructure to where users are based. Control solutions that move networks to the cloud or extend the network to the user are simply extending an already costly solution over additional networks to farther and farther reaches of the internet.

Figure 65:  
Network layer  
connectivity  
requires policy  
enforcement to  
be done using  
legacy network  
controls.



If you share a network context, then anyone can “knock on your ports” just like a person can knock on your door. This is essentially discovering everything on a shared network because the controls are on the network.

It is this knocking process that allows the curious or worse, the malicious, to discover the services, hosts, names, domains, etc., of your enterprise network.

In the houses and streets example, you put locks on your doors (passwords) and put up multiple physical (network) controls like fences, gates, and hedges (firewalls, ACLs, etc.) around your houses or buildings to protect them. You also replace those fences once they get too old. Sure, this makes it harder for bad people to get in, but it still means people can see that a property exists.

Networks are relevant in delivering zero trust application traffic, but they are not the mechanisms where control is enforced. Zero trust controls are not network focused, they are applied regardless of the location of the initiator, or the workload that is being accessed.

This requires two architectural shifts:

### Globally available access controls

Ensure control is available regardless of location and situation. If the initiator requires access, then enforce the control policy.

### Dynamic, granular, and variable controls

Apply controls not only at the application layer, but also specific to each individual initiator and their rights to consume the destination service.

This means it doesn't matter where services are: the controls are applied uniformly. Controls should be independent of the network and be defined for application access. The network is merely a means of transport, thus controls must be applied "on top."

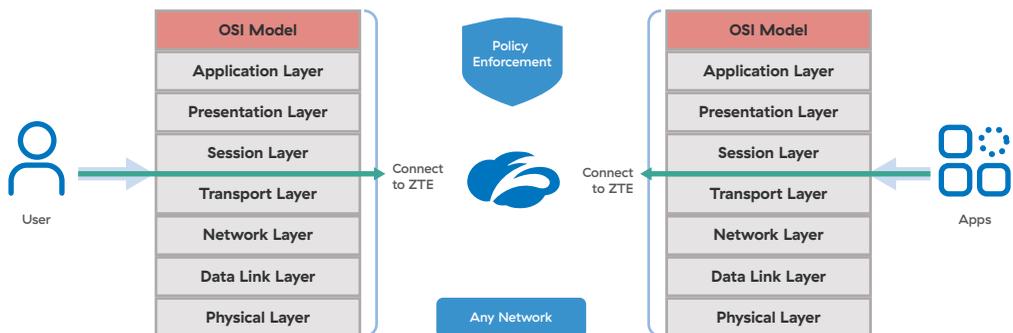


Figure 66: The Zero Trust Exchange overlays a policy enforcement control layer that is abstracted from the network layer.

# How does the Zero Trust Exchange accomplish this?

The previous elements determined whether or not to connect the entity to the desired resource. Element 7 ensures that the connection itself is in fact protected.

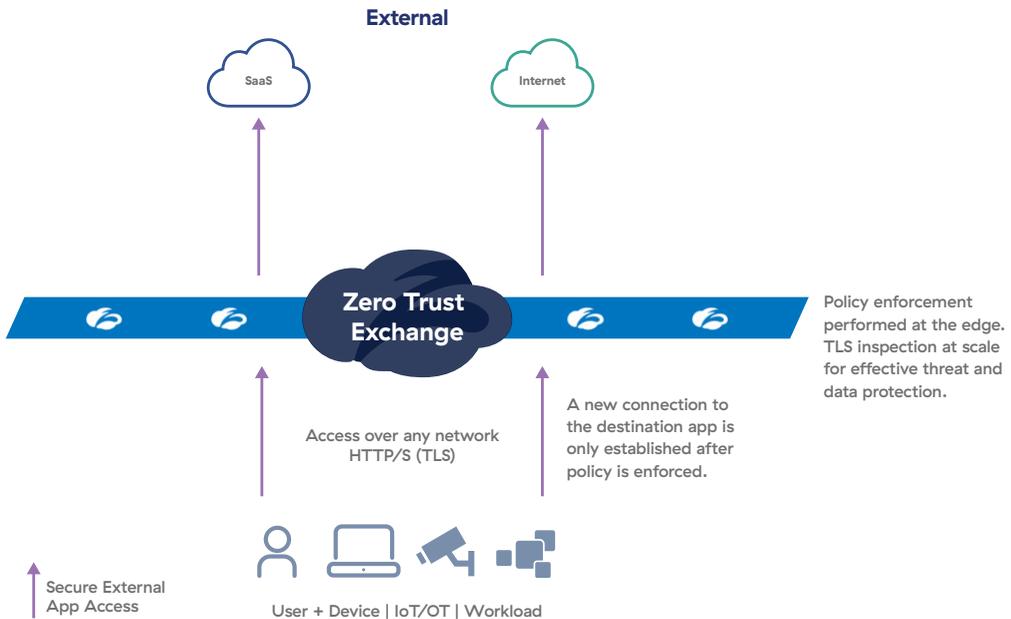


Figure 67: For external apps, connections pass directly from the Zero Trust Exchange to the destination after policy is enforced.

Establishing connectivity from the Zero Trust Exchange to SaaS or internet-hosted applications is relatively straightforward. It is simply an outbound connection that creates the TCP session between the Zero Trust Exchange and destination and completes the user/device, IoT/OT device, or workload-to-application connection.

Note: The destination application path can be selected, or steered, using the Zscaler Internet Access Policy, allowing customers to determine where their traffic will egress to the internet. This helps customers address topics like SourceIP anchoring challenges where internet services only work with geo-controlled IP sources.

## Connecting to the Applications

Internal applications requiring privacy and stronger protection are slightly more complicated; the Zero Trust Exchange leverages App Connectors, which sit adjacent to the application destination. These App Connectors provide the outbound-only connection from the app environment to the broker function within the appropriate Zero Trust Exchange Service Edge.

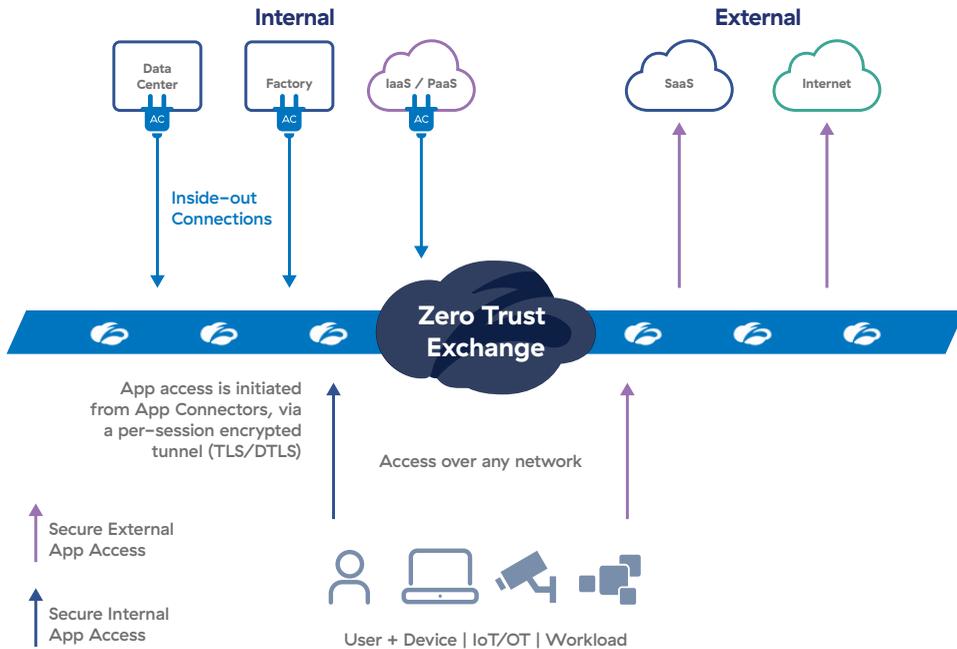


Figure 68: For internal apps, connections are brokered by the Zero Trust Exchange through two outbound tunnels.

Both initiators and App Connectors leverage an encrypted connection outbound to the Zero Trust Exchange. The Zero Trust Exchange then functions as a broker to stitch these connections together and allow users access to the application. This function is similar for users or workloads. This is very important for internal applications, as the main goal of zero trust architecture is to eliminate the attack surface presented by exposed IPs of firewalls and VPNs that protect these internal apps.

The Zero Trust Exchange Service Edge connection, while [generally public](#) and accessible to any authorized entity from the internet, can also be run locally within individual customer locations for seamless deployment of the zero trust controls.

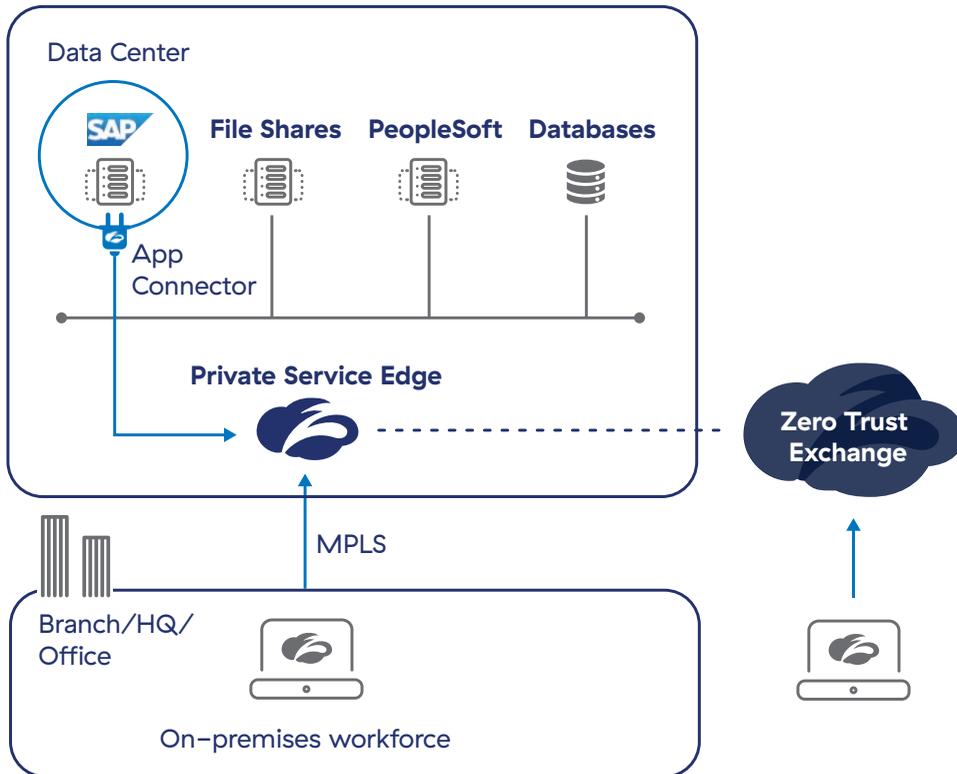


Figure 69: The Private Service Edge, providing local zero trust enforcement inside an enterprise ecosystem.

Local deployments allow for sessions to be controlled within regions or sites to best suit the end customer. These services are similarly broken down by the Internet and Private app paths. [Virtual](#) and [Private](#) Service Edges are available for Internet Application Access and [Private Service Edges](#) are available for Private Application Access.

In the situation where the policy engine defines Browser Isolation as a Conditional Allow path, the Zero Trust Exchange will not allow an initiator to access HTTP-based applications directly. Rather, these isolated access requests will have the workload rendered for the user within the Zero Trust Exchange by streaming pixels. The backend connection, however, remains the same:

### Internet Access

Direct call from the Zero Trust Exchange to the destination application

### Private Access

App Connector establishes the outbound app connection path and delivers user traffic to the application

When connecting to internal applications, the Zero Trust Exchange allows enterprises to be as refined as needed with their connection controls. This is best demonstrated by three examples:

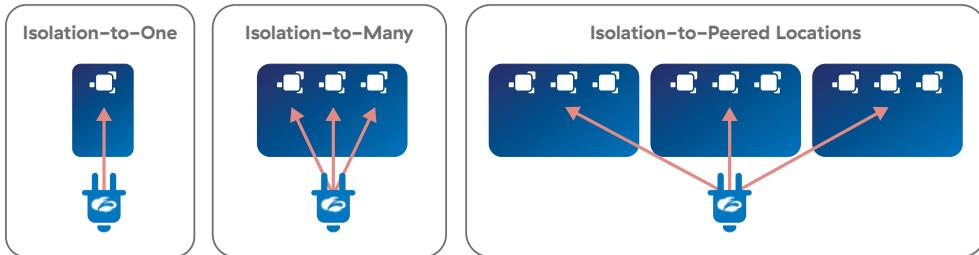


Figure 70: Internal application isolation examples.

## 1. Isolation for one app

Rarely would an enterprise need to isolate at this level of granularity, but allowing a zone of isolation to be one-to-one is done on occasion.

## 2. Isolation for many apps

Specifying access for a segment of sites—whether local sites, application types, or some other group—is a common type of destination isolation for granular controls.

## 3. Isolation for entire peered environments

This strategy is often used during the early discovery phase of a zero trust deployment. Isolation across multiple sites, applications, and infrastructure enables enterprise connectivity and provides insight into which applications are being consumed, even if not as granularly defined as typically desired.

The Zero Trust Exchange consists of three separate planes (control, enforcement, and logging) that deliver functions without relying on specific networks. The Zero Trust Exchange control and enforcement planes are independent of network paths and built to be multi-homed and highly available, so each can run independently of the other.

- The **control plane** is where policy definition and admin are enabled.
- The **enforcement plane** is where Zscaler enforces policy and is [globally distributed](#) to ensure customers receive effective enforcement access. This policy enforcement plane can be extended to internal and on-premises locations and is not limited to remote access use cases.
- The **logging plane** is where configuration takes place and logs are securely streamed to a SIEM/SOAR.

Note: Zscaler visualizes all policy enforcement actions in the [Global Enforcement Dashboard](#).

The control and enforcement planes are independent of the network, but enable any network to operate as a path for access. For example, a network control plane builds and maintains a routing table, whereas the enforcement plane forwards the packets.

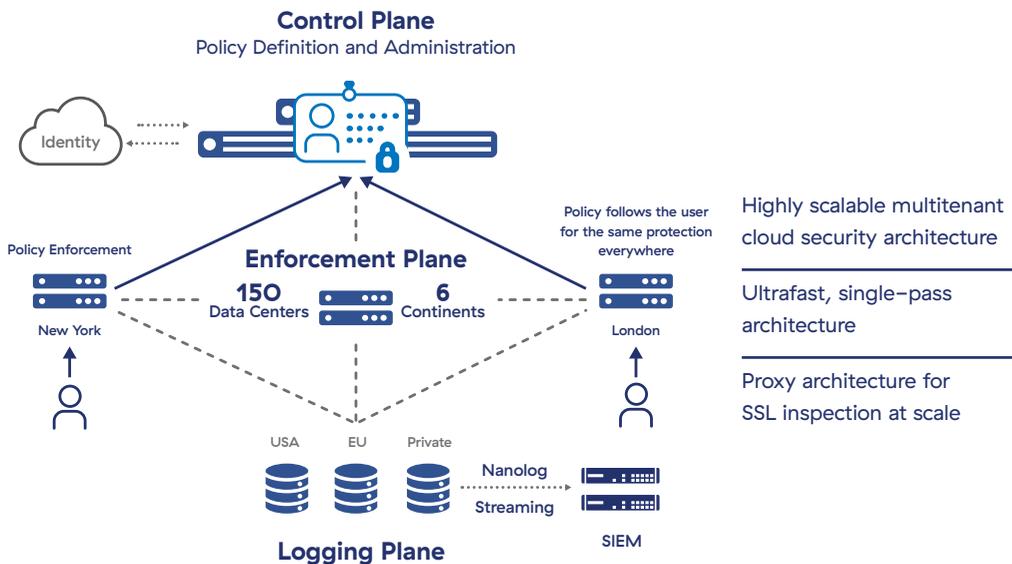


Figure 71: The Zero Trust Exchange's multi-plane design.

By not locking the initiator or the destination to a network, either can live anywhere and still be accessible. This is how the Zero Trust Exchange delivers:

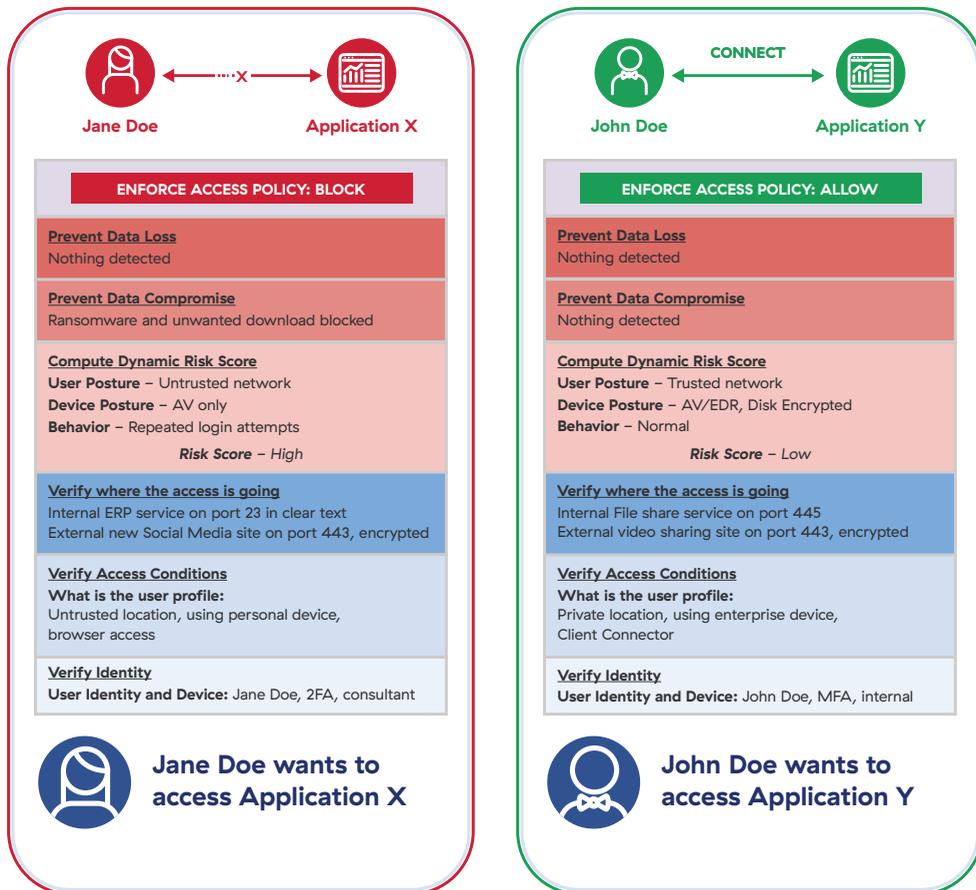
- **A better user experience** — Network-independent access that “just works” where allowed.
- **Advanced protection** — Removes the need for exposed listeners for protection. Instead, all communications are built on outbound paths, including
  - outbound from the end user to the Zero Trust Exchange;
  - outbound from the Zero Trust Exchange to the internet; and
  - outbound from private app environments to the Zero Trust Exchange.

Therefore, users are given access not to networks, but to applications. This access is granted on a per-authorized-session basis so that, before access is granted in this policy phase, all earlier criteria (outlined in Elements 1–6) are verified. If an initiator does not meet the criteria required by policy, they cannot get access, and—for private apps—cannot even see that the requested application exists.

- **Insight and prevention** — Inline inspection allows enterprises to ensure all traffic is
  - protected against malicious sites, malware, etc.;
  - prevented from exposing critical information; and
  - providing insights into the risks, challenges, and patterns facing an enterprise for additional security context.
- **Global availability** — The Zero Trust Exchange is available to customers anywhere in the world via public edges. However, there are circumstances where it may be more efficient to stay within a local network or a specific location. Zscaler is able to deliver this localization scalable to new solutions, such as on-premises, IaaS, and within [mobile edge environments](#). This ensures
  - there is no risk of a lateral threat movement across a WAN;
  - access is unique and assessed per user, per app—each policy control is applied for that session, not for “internet” or “internal”; and
  - Zscaler splits traffic at the client level, not at the cloud level.

# Zero Trust Progress Report

The connection is the final stage of the Zero Trust Exchange for that unique session. Once a given application session is complete, the connection is torn down and, for subsequent requests, the process begins again.



Progress Report 8: Application connection is allowed for John and blocked for Jane, based on the enforced access policy.

# A Fast, Reliable, and Easy- to-Operate Zero Trust Architecture

A zero trust solution must deliver services to an enterprise through a globally distributed platform via a uniform set of policy decision and enforcement edges where any and all enterprise communications are protected. Customers should not simply consider the number of nodes, but rather the number of SLA-guaranteed sites that offer the services they need. A zero trust provider should not provide public PoPs if they cannot guarantee the SLA in that region due to poor peering or for other reasons.

For zero trust architecture to be built properly, the focus must not solely be on security. It must be built in such a way that the user experience is fast and reliable, deployment is as simple as possible, and ongoing operations are streamlined and outage-free. Implementing a zero trust architecture that degrades user experience and is difficult to operate is a recipe for disaster. It will create excuses for affected users to find alternatives, increase technical debt, and impact employee productivity.

To ensure easy operation, speed, and reliability, there are various technical elements to consider when designing a zero trust architecture. A common starting point is agent technology. As many endpoints are already overloaded with security software, zero trust architecture should consider deployment options that can consolidate and remove overlapping agents. As discussed previously, Zscaler employs a single agent, the Client Connector, to enable its zero trust architecture. This same agent forwards traffic to the Zero Trust Exchange for external app protection (data loss and cyber threat protection) as well as brokers a private connection for internal apps, and also provides digital experience monitoring capabilities.

As part of this single-agent architecture, the intelligence to automatically steer internal and external app traffic is important. As these two traffic types take different tunneled paths to either the internal or external application service edges, the intelligence must reside on the endpoint agent to properly identify traffic and send it through the correct tunnel. Failure to have this intelligence on the agent itself requires the painful process of manually configuring client tunnels, which is complex and rife with issues. The Client Connector is designed for this automated path steering and does not require any manual configuration of tunnels.

Additionally, the control of this agent technology, which will live on every employee's endpoint device, must be centralized and allow for the ability to push policy changes and updates easily. The Zscaler Client Connector Portal is built with this centralized control in mind. All bypass policies and forwarding profiles can be managed from here.

## A Fast, Reliable, and Easy-to-Operate Zero Trust Architecture

For flexible deployment options, similar agent technology should be available to deploy at branch sites to enable connection to the Zero Trust Exchange for secure internal and external application transactions initiated by workloads, as well as in the cloud for workloads to have similar connections to the internet or with workloads in other clouds. For these purposes, Zscaler provides the Branch and Cloud Connector components. Similar to the Client Connector, they allow for simple deployment wherever connections need to happen.

While user experience is closely tied to the endpoint agent, the zero trust provider's cloud infrastructure is, not surprisingly, critical for providing a fast and reliable service. Zscaler operates the world's largest security cloud, with over 150 data centers around the world handling approximately 250 billion transactions per day, blocking over 7 billion security violations per day, and experiencing over 200 thousand unique security updates per day (as of August 2022). This cloud is operated with a 99.999% SLA and has direct peering at global internet exchanges to provide sub-millisecond connections to application providers such as Microsoft.



Figure 72: The Zero Trust Exchange is served from 150 service edges around the world and handles over 250 billion transactions per day (as of August 2022).

As part of the security cloud's global presence, consider the cloud's ability to inspect traffic at scale. To maintain minimal latency for inspection of each packet bound for internet and SaaS apps, Zscaler employs a single-pass architecture where the packet is placed into memory once and the inspection services, each with dedicated CPU resources, are able to perform their scans simultaneously.

This avoids the legacy service chaining of these inspections across serialized physical or virtual applications that incur a processing penalty at each hop and run the risk of excess latency imposed on each packet.

Zscaler applies these architectural advantages to newer standards like TLS 1.3, where a true proxy architecture has the advantage of being inline with independent connections to the client and server. Since this allows for the entire object to be reassembled and scanned, advanced threat protection, DLP, and sandboxing can be applied.

Most users will connect to the SSE via a vendor's public service edge. These are full-featured, secure internet gateways and private-application access brokers that provide integrated security. However, situations may arise where the public service edge may not meet requirements, and therefore Zscaler offers private service edge options that are hosted in your own infrastructure. This option extends the public service edge architecture and capabilities to an organization's premises or internal locations and utilizes the same centrally controlled policy as the public service edges.

For secure access to the internet, private service edges can be installed in an organization's data center and are dedicated to its traffic but are managed and maintained by Zscaler with a near-zero touch from the organization. This deployment mode typically benefits organizations that have certain geopolitical requirements or use applications that require that organization's IP address as the source IP address.

## A Fast, Reliable, and Easy-to-Operate Zero Trust Architecture

For internal application access, the private service edge provides similar management of connections between the user and application and applies the same policies as the public service edge, with the service hosted either onsite or in the public cloud but again managed by the SSE vendor. This deployment model allows zero trust within the four walls, which is useful to reduce application latency when an app and user are in the same location (and going to the public service edge would add unnecessary transit). This option also provides a layer of survivability if a connection to the internet is lost. Zscaler distributes images for deployment in enterprise data centers and local private cloud environments.

Both for the public and private service edge infrastructure, the same Zero Trust Exchange provides protection for user-to-app, workload-to-workload (hybrid cloud), workload-to-internet, remote user-to-IoT/OT, and IoT/OT-to-cloud connections.

To ensure optimal performance, the Zero Trust Exchange has its own set of service edges for policy enforcement and does not rely on the content delivery network (CDN) model of connectivity edges from a larger, cloud-based network solely to route or “onramp” your traffic to the central enforcement infrastructure. Such schemes are antithetical to providing highly effective, low latency services.

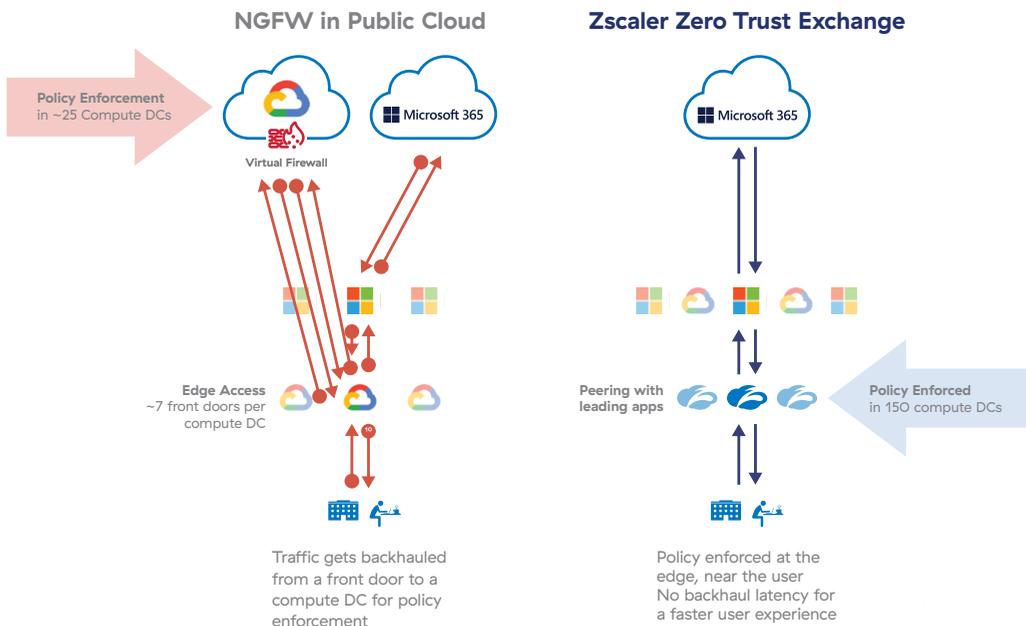


Figure 73: “Onramp service edges” cannot deliver uniform policy enforcement from the edge.

When evaluating solutions, architects should consider the following design foundations, ensuring that policy enforcement edges are:

- Hosted in vital peering locations within carrier-neutral data centers for minimal latency between source and destination. Statistics of availability, routing, and locations should be reviewable in public references like PeeringDB and partner deployments.
- Supported with a valid SLA. This affirms the stability of business functions and indicates the zero trust vendor's ability to deliver a global and available service.
- Capable of deploying privately on a per-customer basis in locations where local conditions require nuanced deployments, such as on-premises or within an edge compute node.
- Able to deliver tenancy protection so customer data privacy is not passed to any other component within the infrastructure and no data is ever stored to disk.
- Providing global-scale protection for all enterprise services once a threat is detected.

The Zero Trust Exchange offers a variety of operational advantages that should be considered as part of the overall solution architecture:

- Operation that can be automated through scripts
- Built-in tools like [speedtest.zscaler.com](https://speedtest.zscaler.com), [ip.zscaler.com](https://ip.zscaler.com), and the Trust portal
- Deployment of an agent through endpoint managers
- App discovery with AI/ML
- Cloud-effect and ongoing cloud updates (vs. hardware appliances)
- Support for managed and unmanaged devices
- Unified policy and centralized control plane
- One-click integration with Microsoft 365
- Ecosystem of partners with robust API integrations

## A Fast, Reliable, and Easy-to-Operate Zero Trust Architecture

An important element of zero trust architecture is the integration of security and digital experience monitoring. The experience of end users and the performance of applications can be monitored and diagnosed with Zscaler Digital Experience (ZDX). ZDX provides digital experience insights to aid in understanding, diagnosing, and improving user experience issues. The ZDX score uses machine learning to help identify performance anomalies and send actionable alerts, with CloudPath analysis that identifies network issues between the user endpoint and their WiFi, ISP, backbone, and the Zscaler service edge.

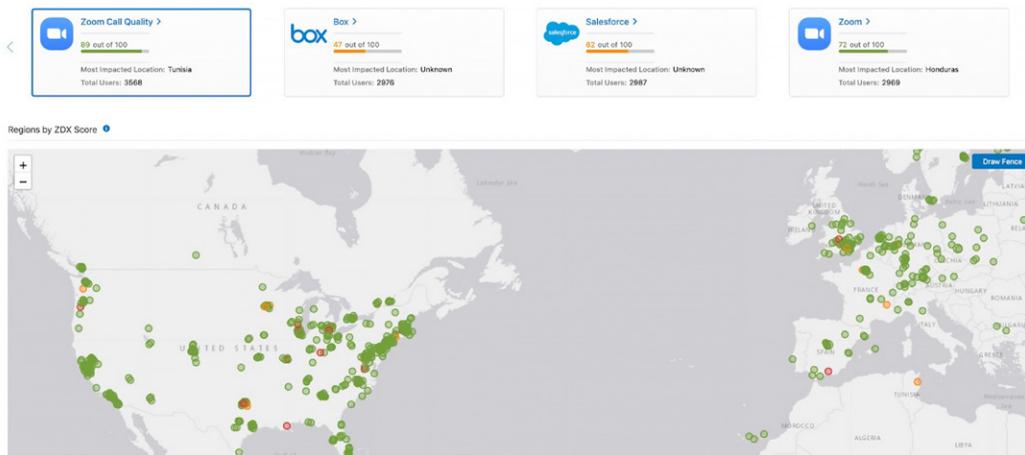


Figure 74: ZDX interface showing the experience of the user across an organization.

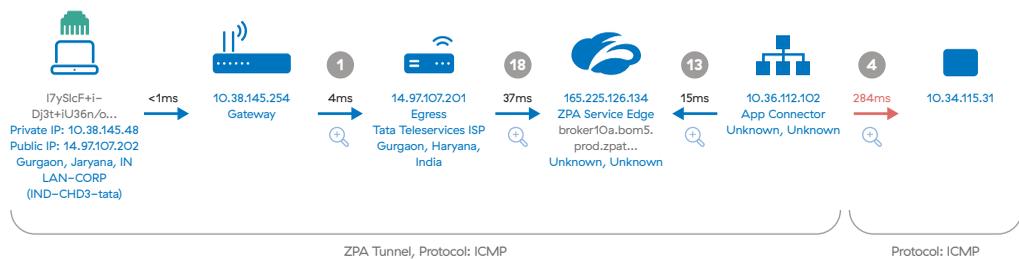
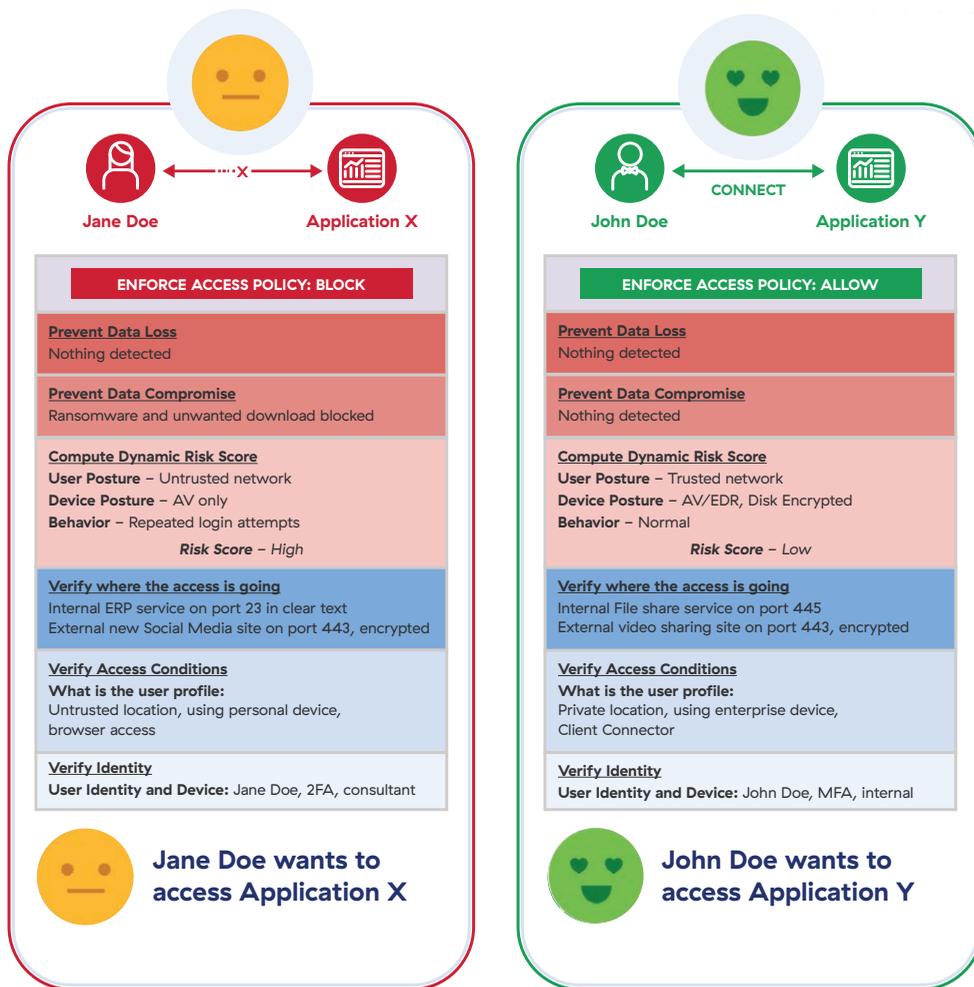


Figure 75: ZDX CloudPath provides hop-by-hop network analysis.

## A Fast, Reliable, and Easy-to-Operate Zero Trust Architecture



*Progress Report 9: Whatever the outcome of the access policy enforcement, a negative user experience is avoided.*

Zero trust vendors must have a demonstrated comprehensive, massive, and resilient cloud platform. Beyond SLAs, the zero trust platform should also provide evidence of scalability, stability, availability, and geographic deployment. To validate this review, consult publicly provided historical data and speak with existing customers to understand their experiences.

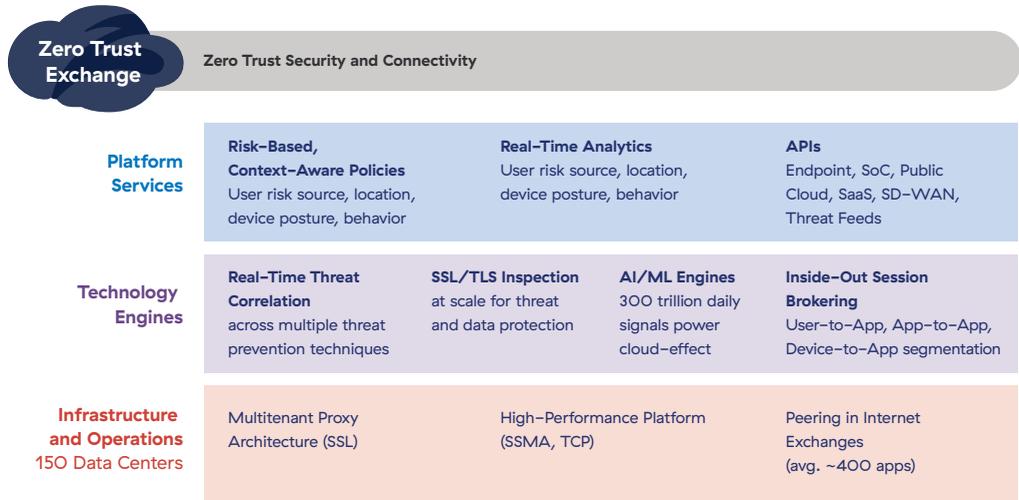


Figure 76: The foundations of the Zscaler Zero Trust Exchange.

# Getting Started with Your Zero Trust Journey

The preceding sections covered seven elements of a highly successful zero trust architecture—when employed successfully, these architectural practices can rectify the inadequacies of legacy network and security architectures.

When a user authenticates with a VPN, the user generally gets full network IP protocol access. Cybercriminals can then exploit this protocol or leverage it for reconnaissance in the attack phase. Attackers can use it to probe networks and data centers or, worse, steer ransomware to additional targets. Users connect to a network using direct IP communication via an IP address, exposing the entire network of listening ports to attackers. An attacker can then use a port scan of subnets to obtain a full list of listening services open on the server.

Connecting users, devices, and workloads to destination services without placing them on the same routable network eliminates the risk of lateral threat movement. Resources sitting behind the Zero Trust Exchange are not discoverable from the internet or corporate network, eliminating the attack surface. All connections are inside-out. No inbound connections to internally managed apps are permitted.

Initiators who are authorized to connect to specific destinations will not be able to learn, connect, or even identify network-level information. Connections to the Zero Trust Exchange are completed only when allowed by policy and deemed safe to be directed to the destination application.

The Zero Trust Exchange is an intelligent switchboard that uses policy to permit connections to destination applications. The Zero Trust Exchange makes the adoption of the often daunting zero trust concepts more feasible in today's world of cloud and mobility. Even when beginning with limited understanding of application usage, the Zero Trust Exchange simplifies the operationalization through intuitive policy frameworks.

To achieve these zero trust benefits, it is important to begin the zero trust journey by answering the following questions:

1. How are my users, IoT/OT devices, and workloads going to connect to the Zero Trust Exchange? How do I leverage the Zscaler Client Connector, Branch Connector, Cloud Connector, or Private Service Edges?
2. How are identities verified? How do I integrate with my IdP, and is it supplying the necessary context?
3. How do I create application policies with user-to-app segmentation rules, while accounting for known versus unknown apps?
4. Which traffic should be decrypted and inspected, and what data loss prevention rules should be set up?
5. What factors should influence the dynamic risk scoring algorithm, and what level of risk am I willing to tolerate?
6. What actions should the Zero Trust Exchange take when making its policy decision, and how do I leverage technologies like Zscaler Browser Isolation?
7. How do I ensure my users are getting a satisfactory user experience, and how do I leverage Zscaler Digital Experience to tell me when they are not?

Successfully embarking on a zero trust journey requires a phased approach.

Begin by securing internet access and phasing out VPN technology for remote users. This often starts with redirecting internet-bound traffic through the Zero Trust Exchange with default filtering and access policies, while also sending critical internal application traffic through the Zero Trust Exchange with a \*.\* access policy to mimic existing VPN access.

This will deliver three distinct early benefits for enterprises:

1. Ensure that applications are accessed in the most direct way by dynamically determining the optimal path to each application in a secure manner. With users no longer on the enterprise network, companies are free to assess which parts of their infrastructure are no longer needed.
2. Accumulate granular application inventory, not limited to IP address, of accessed applications. Each time Zscaler determines the best path for application access, it subsequently documents who accessed which application.
3. Reduction of the attack surface, as key internal applications and the VPN infrastructure are no longer publicly exposed.

Proceed in the phased deployment by defining more protective and controlled policies, which happens with accurate insight into which initiators access which applications. By leveraging the previous output, enterprises can group and organize applications based on functionality, enriching the App Policy definition. Machine learning applied to the accumulated data simplifies this organizational process.

Enterprises can then leverage a similar discovery process to determine which groups of applications are afforded which rights. These groups and rights can then be implemented in the next phase of zero trust deployment.

A simple example would entail documenting the servers, domains, and even IP addresses that make up an enterprise application deployment. This will vary from enterprise to enterprise, but at a high level

- a group definition will be created;
- all addresses (names and IPs) will be added;
- mapping of which users need access to this group will be defined; and
- all others would then have a Conditional Block policy applied.

Definition of these groups and their controls allows enterprises to both determine who can access which service and also how and under which circumstances. This is useful for such purposes as isolating servers and services from any infrastructure, user, application, etc., unless controlled and permitted by the Zero Trust Exchange.

By using the mapped inventory of applications, their groups, and rights, enterprises can then lock down controls governing how workloads access resources. This is a relatively simple process of

- examining the mapped set of groups for hybrid cloud or workload-to-internet communication;
- defining what access is needed for these groups; and
- implementing controls for these workload groups.

Following these steps, all other access would then be restricted and controlled. Extend these capabilities with misconfiguration scanning via CNAPP and finally with workload-to-workload, identity-based microsegmentation.

## Getting Started with Your Zero Trust Journey

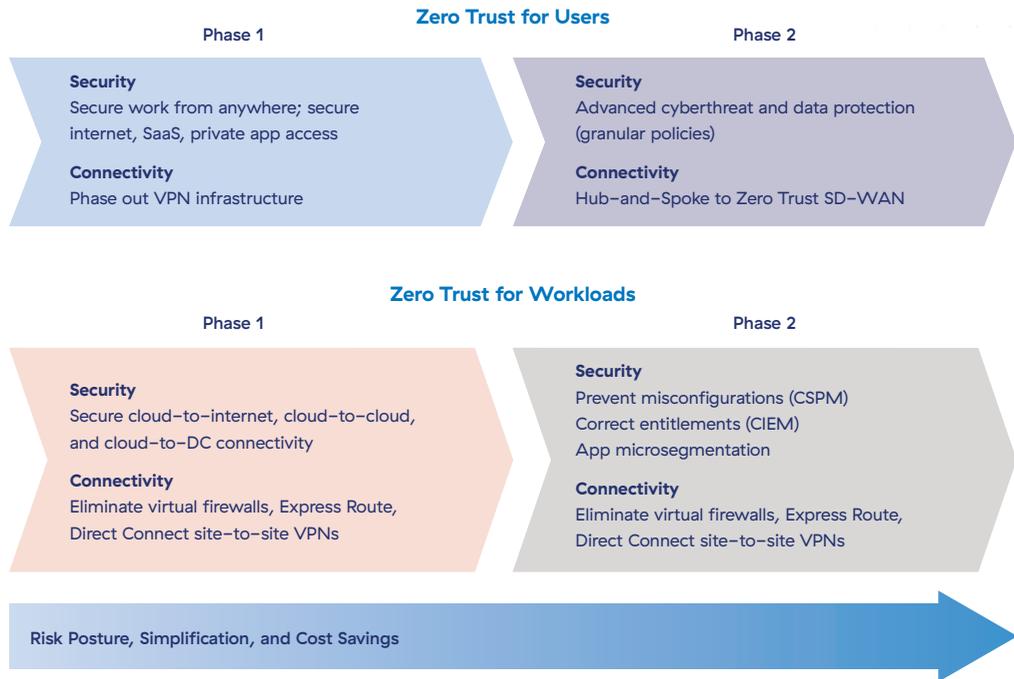


Figure 77: A phased zero trust journey.

As this phased approach is followed, also ensure that the operationalization of zero trust architecture is prioritized:

- Adapt existing processes and culture to best leverage zero trust architecture. Remember that network and security transformation isn't just about technology.
- Evaluate network, security, and endpoint silos, as zero trust will affect all three—ensure all three are aligned.
- Create runbooks that leverage tools provided by the vendor for policy updates, diagnostics, and agent updates.
- Leverage one-click and API integrations with partners to simplify operations.
- Architect a holistic zero trust solution, incorporating all exposed connection types (remote/campus, client/branch/cloud connections) and deploy in a phased manner.
- Create a plan to deploy broadly to take advantage of centralized policy and control planes to simplify operations.
- Deploy proper triage techniques to accurately identify user experience problems (WiFi, ISP, backbone, SSE, endpoint, app, DNS, etc.).

All in all, building a zero trust strategy around Zscaler's Zero Trust Exchange allows for the network and security transformation that ultimately enables digital transformation.

# Appendix 1 – Application Segmentation Primer

Zscaler has a very powerful policy engine for building out granular application access controls as necessary. This can facilitate restricting access, and even visibility, to applications within your ecosystem.

Deploying granular access policy can be complicated in an enterprise given its variables and challenges. That said, Zscaler has established high-level best practices for driving simplicity within policy control and management.

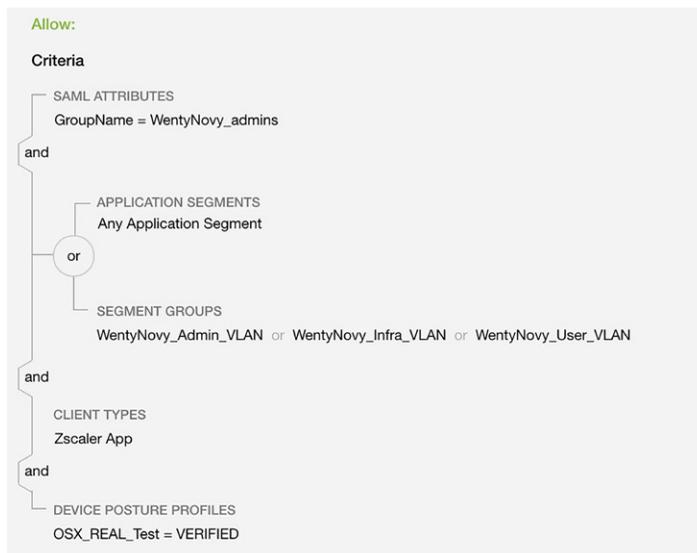
Zscaler has also released a [Reference Architecture for User-to-App Segmentation with ZPA](#) and *The Network Architect's Guide to Adopting a Zero Trust Network Access Service* that explore policy creation and control in much greater depth.

### High-level access policy best practices

Defining a policy is simple and is outlined within the additional documentation. Keep in mind that application policy is defined based on two definitions:

1. The user's presented access criteria, whether that be the SAML attributes assigned to the users within your directory, device context, risk, etc.
2. The application that you are accessing. This is defined, but in actual fact, this is the application context that the user (or the user's device) is calling, e.g., a file share on //fileserver1.company.local.

Here's an example of these two policies as they are applied:



For both applications, customers already have access and control over the data sets.

Let's first focus on user access management:

- The user SAML attributes are sourced from an LDAP/AD/Domain Controller.
- As users are added to a group, each then has the accompanying SAML attribute attached to them.
- This means a large portion of access control is already defined within your internal ecosystem.

Some control mechanism is likely being used to manage the addition and deletion of users and groups, which can also be leveraged in building out a new policy for access. Completing this leverages the prerequisite of building a SAML trust, which likely would have been done during the initial phases of deployment.

So, let's say a customer wants to enable access for its legal team to specific legal applications. In this case (and in most situations), employees are probably already grouped together within a Directory solution under a group or OU. In this case, let's call it Local\_Legal.

Leveraging this existing group membership, when a user logs in to the IdP they are given the SAML Attribute that shows them as part of the group Local\_Legal.

```
SAML ATTRIBUTES
GroupName = WentyNovy_admins
```

Remember, the policy engine respects the SAML attribute. So, simply define a rule that has the user SAML attribute criteria set as "Local\_Legal".

```
SEGMENT GROUPS
WentyNovy_Admin_VLAN or WentyNovy_Infra_VLAN or WentyNovy_User_VLAN
```

Once saved, anyone who logs in (remember, authenticating against the IdP) and is assigned this SAML attribute can then access applications via the rule (or rules) containing this SAML attribute as criteria.

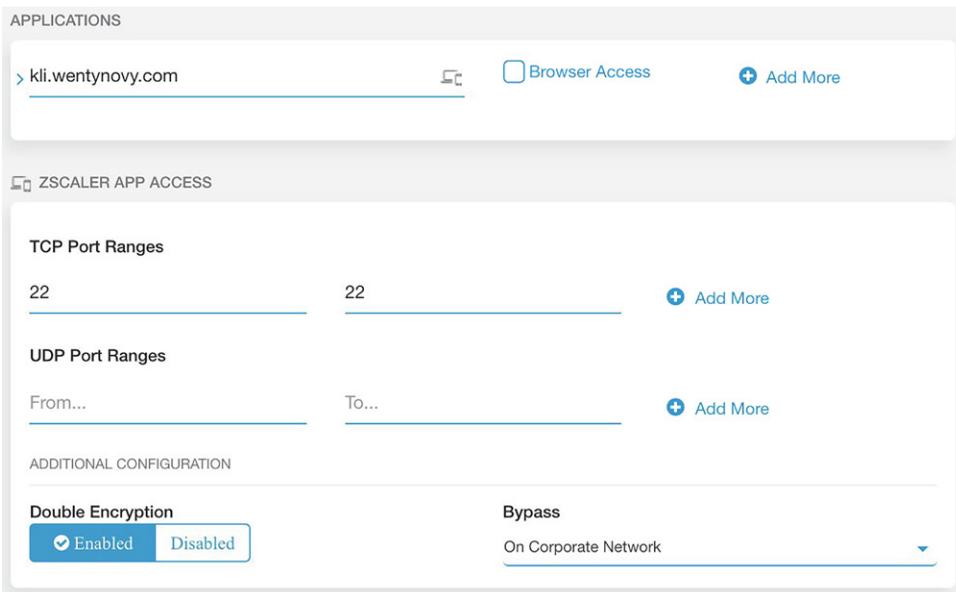
This means that, when new employees join the legal department and are enrolled in the directory and assigned to the Local\_Legal group, they are immediately able to access the necessary apps through Zscaler.

The user has inherited the access based on the attributes that you control in the directory service. You probably already have a process to manage access to these inheritances.

The Zero Trust Exchange applies access rules that you build based on the user attributes and IdP uses.

Next, let's focus on namespace management.

Before diving into details, it is important to remember that the other half of the policy is built on application segments. These are customer-built application definitions. They can be hyperspecific—i.e., a single FQDN to a single port—or broad enough to offer users generic access for visibility and application usage discovery.



In the policy definition defining SAML criteria for access, it's necessary to also define the application segment available to users with this SAML attribute. But let's first look at the application segment definition.

The application segment definition is flexible and will allow you to define a variety of domains, IPs, FQDNs, etc. While it is possible to define each individual application or every IP address in use, this would require constant maintenance of the policy.

Ideally, we ultimately want to define domain space in a way that meets all requirements for that individual application space. So, if everyone on the legal team must be able to access all legal servers (say there are 10) on port 443, why would you build a policy listing each and every FQDN of the legal servers as a separate rule?

Adding each server manually into the policy is acceptable, but it is cumbersome and additional servers would need to be added manually.

Instead, group the servers into a single application segment, then configure a single rule for access to these servers.

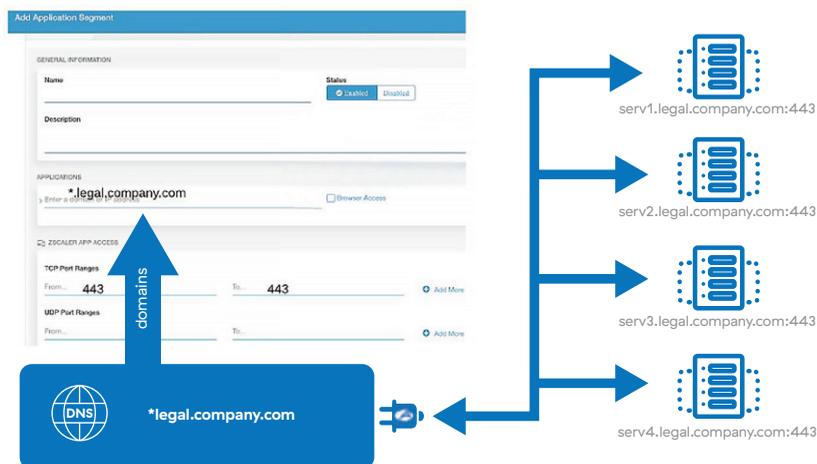
Now let's recall that Zscaler actually validates access based on three factors:

1. What the user (or device) is requesting access to; for example, web browser to legal1.company.local on port 80
2. What the user is allowed access to by policy; is the user allowed access to legal1.company.local:80?
3. If the application access is permitted, is the application reachable at legal1.company.local:80 from the App Connectors?

After answering these questions, define policy in such a way that, while static, allows for dynamic access by managing the local DNS scope for app and server names to match policies.

Ideally, if DNS granularity is established such that a set of legal servers share a namespace subdomain, e.g., \*.legal.company.com, then you would only need your internal servers to have this naming convention.

## Appendix 1 – Application Segmentation Primer



The goal is to simplify policy and management. Ultimately, it's very helpful to ensure that your applications and server names meet a constrained naming convention. In the example above, all legal servers have the name `servX.legal.company.com`, but the idea broadly applies. The key is ensuring the names of all applications of this type fall into the defined namespace.

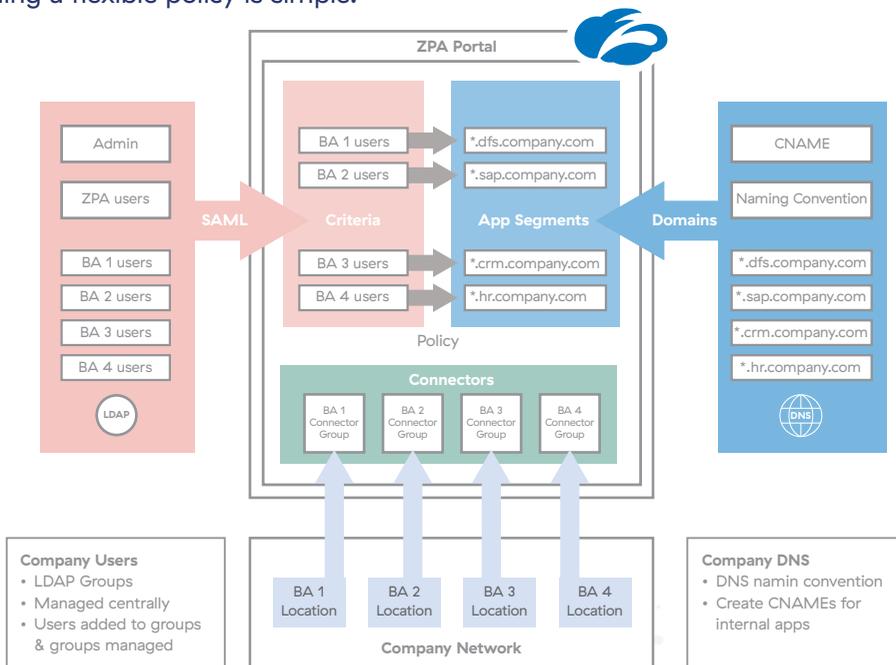
This enables the use of a stable and standardized policy that should be rarely modified, if only to allow exceptions to the rule. Thus, after establishing the policy framework, you would ideally be able to leverage your DNS landscape and existing change management process to add and remove access to allowed users.

Getting to the point where you can roll out this policy does require some specific efforts on the local customer network side.

These efforts include:

1. Defining and refining your domain space, e.g., what network namespaces and subdomains exist, and where they reside within your organization. This DNS namespace can then be used in policy.
2. Ensuring your DNS naming convention is defined and used by the applications in the element. The policy will not permit access if you have defined \*.legal.company.com for access and then your user attempts to access legal1.company.com. Note: For existing apps, you can use CNAMEs on your DNS side as long as the CNAME is called by the client-side.
3. Understanding the necessary ports associated with each application. For example, you can define all of the SSH access under one app segment group with just TCP 22. Similarly, you can define access to all SAP servers on the full set of ports used by SAP.

The above steps should enable access for authorized users to the necessary and allowed applications, without the need to constantly update your policy. After achieving control and understanding of user groups and the application landscape, building a flexible policy is simple.





Rheintec Solutions AG  
Klybeckstrasse 141  
4057 Basel  
Switzerland