Zero Trust Implementation

Zero Trust Training - Training course study guide

cloud security alliance®

ZTT™ Zero Trust Training

The official location for SDP and Zero Trust Working Group is
https://cloudsecurityalliance.org/research/working-groups/zero-trust/

Disclaimer

Cloud Security Alliance designed and created this Zero Trust Training course study guide (the "Work") primarily as an educational resource for security and governance professionals. Cloud Security Alliance makes no claim that use of any of the Work will assure a successful outcome. The Work should not be considered inclusive of all proper information, procedures and tests or exclusive of other information, procedures and tests that are reasonably directed to obtaining the same results. In determining the propriety of any specific information, procedure or test, professionals should apply their own professional judgment to the specific circumstances presented by the particular systems or information technology environment.

Version Number: 20230822

About Cloud Security Alliance

The Cloud Security Alliance℠ (CSA) (www.cloudsecurityalliance.org) is the world's leading organization dedicated to defining and raising awareness of best practices to help ensure a secure cloud computing environment. Cloud Security Alliance harnesses the subject matter expertise of industry practitioners, associations, governments, and its corporate and individual members to offer cloud security-specific research, education, certification, events and products. Cloud Security Alliance activities, knowledge and extensive network benefit the entire community impacted by cloud—from providers and customers, to governments, entrepreneurs and the assurance industry—and provide a forum through which diverse parties can work together to create and maintain a trusted cloud ecosystem.

CSA Address

709 Dupont St.
Bellingham, WA 98225, USA
Phone: +1.360.746.2689
Fax: +1.206.832.3513

Contact us: support@cloudsecurityalliance.org
Website: https://cloudsecurityalliance.org/
Zero Trust Training Page: https://knowledge.cloudsecurityalliance.org/page/zero-trust-training
Zero Trust Advancement Center: https://cloudsecurityalliance.org/zt/
Provide Feedback: support@cloudsecurityalliance.org
CSA Circle Online Community: https://circle.cloudsecurityalliance.org/
Twitter: https://twitter.com/cloudsa
LinkedIn: www.linkedin.com/company/cloud/security/alliance
Facebook: www.facebook.com/csacloudfiles
CSA CloudBytes Channel: http://www.csacloudbytes.com/
CSA Research Channel: https://www.brighttalk.com/channel/16947/
CSA Youtube Channel: https://csaurl.org/youtube
CSA Blog: https://cloudsecurityalliance.org/blog/

Acknowledgments

Dedicated to Juanita Koilpillai, a pioneer in software-defined perimeters whose contributions to the Zero Trust Training and CSA are immeasurable.

The Zero Trust Training was developed with the support of the Cloud Security Alliance Zero Trust Training (ZTT) Expert Group, whose members include volunteers from a wide variety of industries across the globe. Made up of subject matter experts with hands-on experience planning and implementing ZTT, both as cloud service consumers and providers, the ZTT Expert Group includes board members, the technical C-suite, as well as privacy, legal, internal audit, procurement, IT, security and development teams. From cumulative stakeholder input, the ZTT Expert Group established the value proposition, scope, learning objectives, and curriculum of the Zero Trust Training.

To learn more about the Zero Trust Training and ways to get involved please visit: https://cloudsecurityalliance.org/zt/

We would also like to thank our beta testers, who provided valuable feedback on the Zero Trust Training.

Lead Developers:

Clement Betacorne, CISSP, One Step Beyond Group, France
Heinrich Smit, CISSP, CISA, CRISC, Semperis, USA
Mark Schlicting, CISSP, CCSP, BPM, USA
Michael Roza, CPA, CISA, CIA, MBA, Exec MBA, CSA Research Fellow, Consultant, Belgium
Prasad T, OCSP, Head - Information Security, Verse Innovation, India
Shruti Kulkarni, CISA, CRISC, CISSP, CCSK, ITIL v3 Expert, ISO27001 LA, 6point6, UK

Contributing Editors:

AJ (Alexander) Stein, NIST
Ledy Eng, CISSP, CCSP, CCSK, CASP+, CySA+, Security+, Contractor - Undisclosed, Indonesia
Richard Lee, CISSP, CCSP, WCP, MITRE, USA
Robert Morris, CISSP, GDSA, GCIH, MITRE, USA
Roland Kissoon, MBA, MSc, CCSK, CISSP, CRISC, CISA, PMP, Microsoft, USA
Ross Kovelman, Merck

Expert Reviewer:

Abbas Kudrati, Microsoft, Australia
Adish Jain, Policybazaar, India
Amit Butail, Wells Fargo, India
Andy Radle, MITRE, USA
Ashwini Siddhi, Threat Modeling Service Owner, Dell EMC, India
Aunudrei Oliver, CISSP,CCSP,CWNE,CWDP,CCISO,CDPSE,CGEIT, CISCO, USA
Charlie Soto, CISSP, CISM, CDPSE, CIAM, TOGAF, C|CISO, Mandiant Strategic Services (APJ) Now Part of Google Cloud, Thailand
Chris Willman, MITRE, USA
David Skrdla, CamGen Partners/American Fidelity, USA
Donald Byers, Cisco Systems, Inc., USA
Gustavo Vallejo, University in Peru, Peru
Jaye Tillson, CCSK, MBA, Axis Security, UK
John Kindervag, ON2IT Cybersecurity, USA
Karthik Ramamurthy, KNEIP (Deutsche Börse Group), India
Madhav Chablani, India
Matt Lee, CISSP, CCSP, CCSK, CFR, Pax8, USA
Matt Meersman (Dr.), MITRE, USA
Michael Herndon, CCSP, CISSP, CRISC, CGEIT, CIPP/US, CIPT, AWS Certified Solution Architect, Bayer Business Services, USA
Naresh Kurada, Deloitte, Canada
Philip TM Pearson, Aqua Security Software Inc., UK
Reto Kaeserm, Astarios, Switzerland
Roeland van Zeijst, Dutch National Police, Netherlands
Ron Kearns, SCF, CISSP, CCSP, GCSA, Charter Communications, USA
Ron Martin (Dr.), PhD, CPP, Capitol Technology University, USA
Sakthiswaran Rangaraju, Pure Storage, USA
Shain Singh, F5, Australia
Shinesa Cambric, Microsoft, USA
Sky Hackett, CISSP, CISA, Amazon Web Services, USA
Vani Murthy, CISSP, CDPSE, CCSK, CRISC, PMP, ITIL, MBA, MS, Akamai Technologies, USA

CSA Staff:

Adriano Sverko, Cloud Security Alliance, USA
Anna Campbell Schorr, MBA, CCSK, Cloud Security Alliance, USA
Chandler Curran, Cloud Security Alliance, USA
Daniele Catteddu, CISM, Cloud Security Alliance, Italy
Hannah Rock, Cloud Security Alliance, USA
Noelle Sheck, Cloud Security Alliance, USA
Stephen Smith, Cloud Security Alliance, USA

Table of Contents

List of Figures

Course Introduction

Welcome to *Zero Trust Implementation* by Cloud Security Alliance (CSA). This training module is part of a larger series titled Zero Trust Training (ZTT). It builds upon and extends the concepts discussed in the CSA *Zero Trust Planning*[1] and *Introduction to Zero Trust Architecture*[2] courses. In this course, learners get an in-depth look at the crucial facets of Zero Trust (ZT) implementation, from creating project kick-off documents and disaster planning, to setting up the network environment, deploying agents to devices, and adding automation.

Course Structure

This course consists of four units, each geared towards gaining increased competency in the following topics:

1. Continuing the ZT journey
2. ZT project implementation considerations
3. Implementation preparation activities
4. ZT target architecture implementation

Course Learning Objectives

After completing this course, learners will be able to:

- Identify the assumptions and considerations for continuing the ZT journey
- Explain the main ZT project implementation preparatory activities
- Outline Zero Trust Architecture (ZTA) implementation steps
- Leverage ZT pillars and cross-cutting capabilities to define and prioritize implementation tasks
- Visualize and document security workflow architecture using transaction flow diagrams and tables
- Design testing procedures that can be repeated and generate audit trails
- Define success criteria and review the success of ZT implementation

1 Continuing the ZT Journey

Before we jump into the content of the *Zero Trust Implementation* course, let's recap a few key points that we already addressed in the *Zero Trust Planning*[3] module. The ZT project plan is the roadmap that serves as the team's checklist to implement this plan.

[1] https://knowledge.cloudsecurityalliance.org/zero-trust-planning
[2] https://knowledge.cloudsecurityalliance.org/introduction-to-zero-trust-architecture
[3] https://knowledge.cloudsecurityalliance.org/zero-trust-planning

The *Zero Trust Planning* module covered:

- Starting the ZT journey
- Planning considerations
- Scope, priority, and business case
- Gap analysis
- Defining the protect and attack surfaces
- Documenting transaction flows
- Defining ZT policies
- Developing a target architecture

Before diving into implementation of ZT, it's important to set the stage and make some general assumptions, which we cover in the next section. This is due to the varying types of industries and companies that exist; there isn't enough room in this course to delve into all the specifics for each industry, such as healthcare, finance, or energy.

1.1 Training Assumptions

This training assumes that the learner reviewed and understood the preceding Zero Trust Trainings (ZTT): *Introduction to Zero Trust Architecture* and *Zero Trust Planning*. Together, they form a body of work, including defining ZTA, ZT pillars (Identity, Devices, Networks, Applications and Workloads, and Data), and ZT cross-cutting capabilities (Visibility and Analytics, Automation and Orchestration, and Governance). For this training, our ZT implementation is designed around the ZT pillars and ZT cross-cutting capabilities, defined in the Cybersecurity and Infrastructure Security Agency (CISA) *Zero Trust Maturity Model*[4].

Additionally, this training assumes the student knows and understands basic software project management and can create a project plan for implementation. Let's quickly review the main ZT project management implementation steps we have already covered:

1. Project organization (covered in *Zero Trust Planning*): A company defines the protect surfaces and priorities, determines what objectives must be met and by whom, and identifies the steering committee.
2. Project design (covered in *Zero Trust Planning*): The project team maps the transaction flow, defines the ZT policies, and designs the ZT environment.
3. Implementation (covered in *Zero Trust Implementation*): During ZT implementation, the solution is set up and documented. Frequent status updates are needed for the project manager and the steering committee to ensure the project is on time and within budget. During this phase, the security team also creates a plan to monitor and maintain the ZT policies and network.
4. Testing (covered in *Zero Trust Implementation*): After implementation, various types of tests are run to prove acceptance criteria are met. These can be classified as systems readiness testing (SRT) and operational readiness testing (ORT).

Lastly, given the comprehensive, enterprise-wide scope of ZT, implementations are usually

[4] CISA. (2023). Zero Trust Maturity Model (Version 2.0).

incremental and iterative (as opposed to entirely new or cut-over implementations). In alignment with project management practices, this course refers to a single iteration of implementation as a **project**, and a collection of implementation projects pertaining to the ZTA goal and scope as a **program**.

The following unit will discuss the ZT project implementation tasks that should be considered before beginning your implementation preparation activities.

2 ZT Project Implementation Considerations

Before implementing ZT, a list of tasks and requirements should be considered:

- The gap analysis report should be consolidated and approved by stakeholders.
- Organizational security policies must include ZT-related security objectives.
- Migration from existing architecture to ZTA requirements needs to be addressed.
- Stakeholders must determine what part of ZT implementation will be done in-house.

2.1 Gap Analysis Report[5]

When you are ready to implement ZT, let's revisit the gap analysis your organization put together during the planning phase (see *ZT Planning*). Simply put, a gap analysis looks at what you have and compares it to what you need, reminding you to focus on each individual protect surface. By looking at each protect surface and what needs to be done, you can prioritize accordingly and start with your ZTA implementation–one protect surface at a time. This makes implementing ZT much more manageable than trying to do it all at once.

The gap analysis report identifies the steps needed to build a target ZTA, which should be prioritized and agreed upon by stakeholders who must coordinate their alignment with the ZT plan and its associated pillars and cross-cutting capabilities. For instance, those concerned with the Identity pillar might identify a security control missing from authentication and opt to introduce multi-factor authentication (MFA) as an appropriate measure to bridge the gap in authentication requirements. In a later section, we delve further into the details of ZT pillars and cross-cutting capabilities.

2.2 Aligning Information Security Policies with ZT

It is important to understand how information security policies connect with ZT principles. ZT is a concept that helps organizations strengthen their security measures. When you align ZT with information security policies, you need to consider the risks that ZT addresses and the security requirements of its core pillars. By considering these factors, you can develop stronger policies that enhance your organization's security posture.

[5] Learn more about "Gap Analysis" here: https://knowledge.cloudsecurityalliance.org/zero-trust-planning

Let's take the example of the ZT Identity pillar. To ensure alignment, you should examine your organization's identity and access management policies, procedures, and processes. Check if all the necessary elements, like the access control review procedure, are in place. If any of these elements are missing, or if ZT requires additional procedures, you may need to create new ones or make changes to existing ones.

2.3 Migration From Existing Architectures to ZTA

To effectively implement a ZTA, it's crucial to clearly understand the project scope and communicate it effectively with your team. This includes determining whether we are implementing an entirely new architecture or migrating from an existing one.

In this training, we assume that ZTA will be implemented in an already-existing environment with controls, whether on-premises, hybrid, or in a cloud-only scenario. To successfully achieve the desired ZTA, assessing the impacts and requirements of the technology involved is essential. When evaluating the architecture for any technology gaps, the team should consider the following:

> - Weighing the benefits and costs of introducing new technologies versus collaborating with existing vendors to enhance their product or service capabilities
> - Finding ways to simplify the environment if it becomes overly complex, aiming for a more streamlined approach
> - Moving beyond simply replacing technology, but instead advancing capabilities in a manner that aligns with the organization's goals and growth path

By considering these factors, you'll be better equipped to make informed decisions, ensure a successful ZTA implementation, and avoid unnecessary complexity.

Similar considerations apply in IT environments that have not yet implemented ZT; however, these systems have fewer dependencies on existing business systems and can be implemented more quickly. When the ZT system or modifications are operational, existing (and now redundant) systems should be decommissioned. This training applies to both situations.

2.4 Managed Service & In-House Implementation

ZTA is a combination of in-house and managed services. In-house implementation can be achieved internally by leveraging teams, such as InfoSec, identity and access management (IAM), and infrastructure, with possible contributions from specialized implementation consultants. A managed services approach includes various vendors that can provide solutions designed to support ZT strategies. While your team may strive to keep the implementation of ZTA in-house, there will always be a level of shared responsibility that needs to be orchestrated and considered.

To choose the best method, the following list identifies some considerations for ZTA implementation planners:

- Cost/benefit
- Capability
- Resource availability and your organization's skill-set availability
- Support
- Shared responsibility
- Policy
- Proof of concept and business proposal

3 Implementation Preparation Activities

Although we previously discussed implementation preparation in Zero Trust Planning, we summarize some essential kick-off activities here:

- Define ZT project deliverables
- Communicate ZT changes to users
- Create an implementation checklist

3.1 Defining ZT Project Deliverables

Prior to starting the ZT implementation, the overarching ZT project team must build some common deliverables that apply across the entire organization (or in-scope target architectures, if the scope is narrower than organization-wide) for each ZT pillar. By doing this with an agile approach, you can work quickly and efficiently, especially in areas that incorporate new automation or involve the development team. Due to the high likelihood that third-party stakeholders will be involved, use waterfall-based project planning models to help you develop milestones, where partial payments can be made for the percentage of work completed. Ideally, all pillars and cross-cutting capabilities should be worked on simultaneously while following the five-step process[6] outlined in the *Zero Trust Planning* training:

- Define the protect surface
- Map the transaction flows
- Build a ZTA
- Create a ZT policy
- Monitor and maintain the network

[6] NSTAC. (2022). NSTAC Report to the President on *Zero Trust and Trusted Identity Management*.

3.2 Communicate ZT Change to Users

All-hands meetings should be held on a regular cadence to ensure everyone is aware of the collective progress when it comes to implementing ZT pillars. Moving forward, ensure users are informed about upcoming operational changes, such as bringing in a new external consultant or introducing a cloud-based service. Communicating the exposed assets and their priority level will help with the successful adaptation of the ZT solution.

Changes to workflows should be communicated early, which has these and other benefits:

- Teams can plan for any adoption impacts
- Implementation teams can reach out to the group to ensure that testing covers all use cases
- Impact on infrastructure access during vital deliveries or engineering cycles can be minimized

3.3 Create an Implementation Checklist

Before kicking off the implementation, create a checklist of milestone activities that maps, where appropriate, the changes required. The list should include changes that will need to be made to:

- Organization's governance
- Compliance
- Risk management
- Operations and maintenance
- Visibility and analytics
- Incident management
- Change management
- Vulnerability and patch management
- Problem management
- Business continuity planning (BCP) and disaster recovery (DR)
- Training and awareness

3.3.1 Organization's Governance

ZT will likely require an update to the organization's governance approach and organizational policies, procedures, guidelines, and security controls. As the organization implements a ZTA, governance practices will guide the implementation functions, activities, and outcomes. The organization's senior leadership will rely on the technical expertise of functional technicians to develop and implement security controls that involve input validation, session management, and password storage. Integration examples of these controls are authentication, authorization, cryptography, input validation, output encoding, auditing and logging, and monitoring and alerting. Organizational governance assures sound stewardship of resources throughout the implementation process.

3.3.2 Compliance

When implementing ZT, it's crucial to remember that existing change control processes, compliance, and auditing requirements are followed, even if changes are needed. Whenever an information system within the scope of ZT undergoes changes, it's essential to follow established change control processes to ensure compliance.

As you keep working on implementing and refining your ZT approach, brace yourself for future laws, standards, rules, and regulations likely to impose more stringent requirements on ZT. Keep a keen eye out for changes in the compliance landscape as governments and organizations increase their insistence on security.

3.3.3 Risk Management

The implementation of a ZT strategy can change an organization's risk posture and risk management approach. To respond quickly to changes in the risk landscape, organizations must have a culture of continuous risk evaluation and policy adjustment. This means that the existing risk analysis and assessment process should include the following elements:

- An assessment frequency that allows for the rapid identification of new risks and new threats
- Metrics (tailored to the organization's networking trends), which are easy to spot in reports and monitoring tools, and that should also be able to send out alerts in extreme cases
- Data from various sources should be pooled together for more useful analysis

3.3.4 Operational Requirements

Operational or business-as-usual requirements promulgated by one or several departments will impose conditions or restrictions on ZT. To ensure that the ZT implementation stays within budget and on schedule, it is important to be prepared by identifying these processes and requirements before the start of an implementation. For example, integration between ZT automation and a configuration database must be agreed-upon and then installed and tested to ensure the operational readiness of agents or API calls.

3.3.5 Visibility & Analytics Integration

Log management and monitoring of cyber events drive visibility, which in turn supplies analytics that 'inform policy decisions, facilitate response activities and build a risk profile to develop proactive security measures[7]. For logs to be useful and to provide value when monitored, the following needs to be identified:

- Log scoping: for example, domain name system (DNS) logs, network address translation (NAT) logs, and intrusion detection systems/intrusion prevention systems (IDS/IPS)
- Log sources: for example, the identity provider (IdP), policy enforcement point/policy decision point (PEP/PDP)

[7] CISA. (2023). Zero Trust Maturity Model (Version 2.0). April 2023, page 11.

- Security events that need to be monitored: for example, failed authentication requests
- Anomalies that need to be detected: for example, five failed authentications within five minutes
- Correlation rules to identify threats or potential threats: for example, an identity that drops a production database is deleted
- Dashboards for visualization of logs: for example, a dashboard that shows privileged activities carried out by all administrators
- Monitoring: for example, continuous monitoring of any network connections made to a known command-and-control
- Alerting: for example, using emails, SMS, security information and event management (SIEM), or security orchestration, automation, and response (SOAR)

The above planning and design ensures that your implementation team leverages all log sources, collects and sends security events to SIEM, and identifies SIEM interfaces (e.g., add-on applications, API calls, and event collectors). These activities prevent mishaps caused by implementers scrambling to get these interfaces together during the implementation phase.

3.3.6 Vulnerability Scanning & Patch Management

While vulnerability scanning and patch management are standard IT practices, ZT acts as a control gate to ensure all systems are patched and business operations can only be performed on a patched system. Identifying the vulnerability scanning and patching requirements for components, such as IdP or PEP, of ZT implementation ensures that implementation services become part of the existing patching process that the organization has. By identifying vulnerability checks and tests, wherever possible, your organization can continue to work quickly and harvest all the benefits of technology without leaving the organization vulnerable to a hack. By identifying patching requirements, you can build patching checkpoints that monitor for unpatched or vulnerable software components within your environment and ensure all components are kept up to date. It may also assess the current security status of an operating system or application, identifying any available patches and upgrades, testing these to ensure compatibility with existing applications, implementing necessary updates, and monitoring for successful patch installation.

The absence of vulnerability scanning and patch management may defeat the very purpose of implementing ZT because a vulnerability may diminish the objective of authentication before authorization.

3.3.7 Change Management Process

ZT needs to be included in the change management process. During this activity, it is important to design and track separate workflows for:

- A service request
- A change request

For example, the onboarding of a device can be treated as a service request. However, adding a policy to the PEP, also known as a gateway, may be treated as a change request. Suppose a change

request is needed because an unforeseen gap has been found during the ZT implementation. This change may require an emergency budget, developer- or engineering-based redesign, or consultation from a third party. This type of change may need modifications to the code, the network, or the environment and will need to be queued into the problem management process.

3.3.8 Problem Management Process

ZT should be integrated into the organization's problem management process to address any recurring incidents, future incidents, and methods for reducing the impact of incidents that cannot be prevented. Problem management processes should cover problem detection, problem logging, error control, and root cause analysis, to name a few.

3.3.9 Incident Management

Incident management is focused on addressing incidents in real-time and may need to be revised based on the changes the ZT approach will bring to the organization. As part of this, the ZT implementation team needs to define what constitutes a security incident in a ZT solution and what constitutes a significant security incident. The team also needs to identify the list of people that must be contacted in the event of a significant incident, along with their contact info.

3.3.10 Business Continuity Planning & Disaster Recovery

BCP and DR are important processes that should be aligned with ZT implementations. As ZT deals with access to resources to carry out IT activities, existing access processes may not work as expected due to ZT implementation activities. Business continuity activities should be planned during implementation such that disaster recovery can be addressed during this phase.

3.3.11 Training & Awareness Programs

New technologies, architectures, and solutions introduce new workflows and ways of working. While preparing for ZTA implementation, it is important to keep training requirements in focus. An implementation solution will be productive and effective if assigned teams are comfortable using it, which comes from regular training and awareness-building.

To do this, list all ZT implementation training and awareness programs that need to be scheduled. Also, identify the frequency at which the programs need to be scheduled. This will help organize and budget training. While the details will vary, depending on what the implementation entails and the nature of your organization, training requirements may be categorized into two groups:

- Business and operational goals
- Technical goals

Business and operational goals include items, such as:

- Training and awareness that publicizes the ZT implementation business goals
- Building awareness about responsibilities, such as the budget owner, asset and process owner, legal team, architects, security team, and IT team
- Training architects
- Training support personnel
- Training for risk managers and compliance officers
- Training to understand new forthcoming operational processes, such as change management, incident management, BCP drills, and so on

Technical goals include items, such as:

- Enrolling endpoints to ZT architecture
- Creating and maintaining policies
- PEP and PDP maintenance
- Creating and maintaining micro-segmentation
- Maintaining identity provider services and databases by working with identity creation, access control reviews, and suspension of identities belonging to offboarded employees

Feedback should be obtained during these training sessions so that updates can be made to the training and awareness program.

4 ZT Target Architecture Implementation

We have just explored a series of milestone activities that must be reviewed and considered to support a ZT target architecture implementation. With the above objectives in mind, this section will explore the following implementation themes:

- ZT helps in reducing access related compromises by authenticating and validating the access request.
- Organizations must have the right personnel and systems to monitor suspicious activity and policy violations.
- Integrate ZT testing efforts into the cybersecurity program and promote collaboration between departments to ensure the organization maintains secure operations.
- Prioritize risk management, review transaction flows, continual improvement, and project closure.

Imagine a real-life scenario: We have planned and prepared the following ZT reference architecture, as shown in the diagram below.[8]

[8] Note: Please refer to CSA trainings: Introduction to Zero Trust Architecture and Zero Trust Planning for a review of the components depicted in this diagram and their respective functions. These components are also defined in CSA's Cloud Security Glossary.

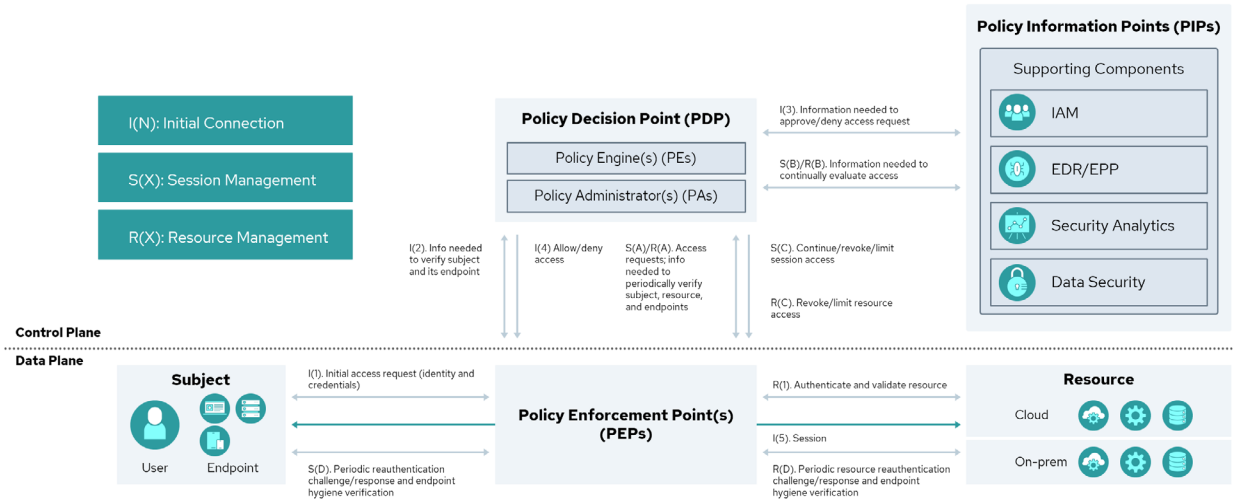

Figure 1: *General ZTA Reference Architecture*[9]

This diagram captures the flow of operations when the subject accesses a resource. When a valid subject wants access to a resource, the request is initiated from the endpoint to the PEP in the following steps:

1. An initial client authentication request is sent from an endpoint to the PEP.
   NOTE: The operation may repeat based on the identified ZT requirements.
2. Information required to verify the subject and endpoint is collected by the PEP and shared with the PDP.
3. The PDP validates the device and subject authentication. At the final, advanced stages of ZTA implementation, this point can be integrated with various policy information point (PIP) solutions and technologies. If the validation succeeds, the PDP decides on the type or level of authorization needed.
4. The PDP informs the PEP about the authentication status for the connection and authorization details if obtained.
5. The session established to check a user's credentials or endpoint is now terminated. The established sessions will undergo periodic validations and terminate per the predefined rules.

It is important to note that this flow of operations is separated by the control and data plane. The control plane decides the path for sending packets or frames and directs how these packets should be forwarded. The data plane is where the action takes place; it's all about the functions and processes that move those packets from one interface to another.

Having discussed the ZTA reference architecture above, let us dive into the ZT pillars and cross-cutting capabilities with the intent of studying their attributes in more detail. This will help us better design the ZT implementation across all pillars and cross-cutting capabilities.

[9] Figure adapted from: NIST. (2022). Implementing a Zero Trust Architecture (SP 1800-35B). Second preliminary draft.

4.1 Zero Trust Pillars & Cross-Cutting Capabilities

One way ZTA implementation can be coordinated is within and across the ZT pillars and cross-cutting capabilities. As ZT functions are added, each pillar should mature and evolve from the traditional level at the start to the optimal level as you near the final implementation phases.

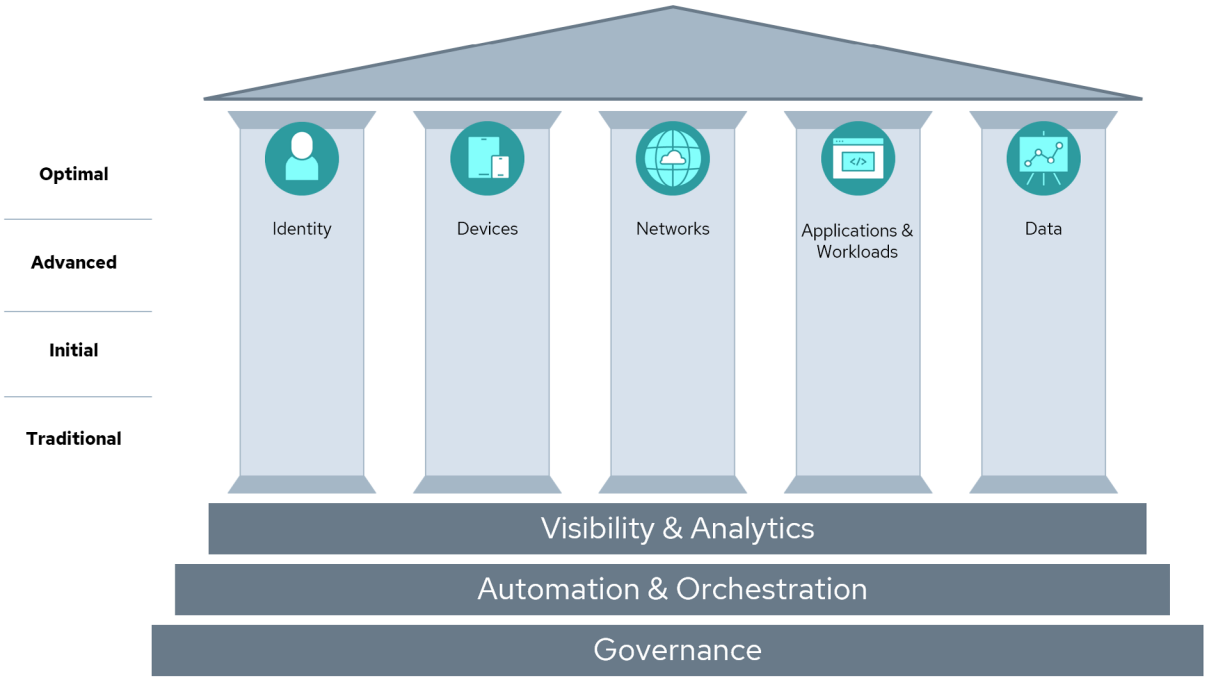

Figure 2: *Zero Trust Maturity Evolution*[10]

As displayed above, the five ZT pillars are:

- Identity
- Devices
- Networks
- Applications and Workloads
- Data

Cutting across these pillars are capabilities, referred to as cross-cutting capabilities, that improve at every maturity level and interact with each pillar:

- Visibility and Analytics, primarily by aggregating output
- Automation and Orchestration
- Governance, by introducing governance and compliance software

In some diagrams (i.e., from the US Department of Defense[11]), the cross-cutting capabilities (Visibility and Analytics, Automation and Orchestration, and Governance), are depicted as foundational pillars. This representation emphasizes the significance of incorporating these capabilities into the implementation process as they assist in defining objectives for the five pillars.

[10] Figure adapted from: CISA (2023). Zero Trust Maturity Model (Version 2.0).
[11] U.S. Department of Defense. (2022). DoD Zero Trust Strategy.

Before covering each pillar and cross-cutting capability in depth, the relevant attributes for each pillar and cross-cutting capability should be identified and defined, as noted in the following table.

Pillar	Pillar	Attribute	Notes
P	Identity	• Identify all the identities that are required to be part of the ZTA • Identify the identities that requireaccess to the protect surface • Define permissions for identities according to least privilege using custom roles (a group of individual permissions) and assigned to a role (group), except in situations where access may be restricted to a single service account or other non-human identity • Monitor modification/drift for the membership in the role/group as well as the individual permissions associated in the custom role	• This should be treated as infrastructure components
P	Devices	• Identify all devices that need to be enrolled • Identify business & security applications running on devices	
P	Networks	• Identify the necessary macro-segmentation in data centers& micro-segementation	Micro-segmentation between hosts within the: • VLAN • VPC • CNet
P	Application & Workloads	• Identify your applications and workloads that exist in your on-prem or cloud infrastructure	
P	Data	• Identify data sources that are part of your protect surface • Identify transaction flows	
C	Visibility & Analytics	• Identify the registry (logs) repository in all entities (users/ identities, devices/endpoints, network and environment, and applications and workload) • Identify the external data that you need to enrich the visibility • Identify the parameters you need to monitor the performance, behavior, and activity of the ZT deployment	

C	Automation & Orchestration	• Identify the security operations to perform when a policy allows or denies actions • Identify the standards that support the security and non-security technologies to communicate with others • Identify the conditions and technologies of the security playbooks	
C	Governance	• Identify the governance structure that ZTrequires	
		• Policies and procedures	• Organizational security policies: • Identify the information security policies that ZT deployment needs to adhere to
		• Controls	• Identify the security controls that need to be applied to the devices, network, applications and workloads, and data • This should not be confused with the transaction flow controls • Identify the controls to be implemented across the user agent, such as authentication, authorization and various other aspects • Identify the rule-based access policies that are part of the PDP, also referenced as the controller, and have been identified in the planning session
		• Risk management	• Identify the risk managment requirements
		• Compliance	• Identify the risk compliance requirements

Table 1: *Implementing ZT Across Pillars & Cross Capabilities*

Now that we have seen overviews and attribute summaries of the pillars and cross-cutting capabilities, the remainder of this unit will discuss implementation-specific details and considerations that tie these to executing the implementation.

4.1.1 Identity

Identity is the pillar involving authentication and authorization, including privileged access management (PAM). Effective use of identity as a data source will include centralized directories (as

well as related onboarding strategies) and federation between enterprises. The rules and validations related to identity are implemented at the PDP, where a predefined IdP provides user management and other validations, such as risk assessments and device posture validations. The IdP can be within the enterprise or external to the enterprise (e.g., as a SaaS application). When the IdP is within the organization's network, the PDP can reach the IdP through the PEP.

4.1.1.1 PDP Identity

PDP users that monitor and manage the PDP–we can call them PDP admins–must create policies and perform maintenance on the ZTA at the control plane. To reduce the impact radius of potential cyber-attacks, each of these PDP administrators, and any other user with elevated permissions, needs at least two identity profiles:

One for their day-to-day activities, such as reading emails, surfing the web, or using the ticketing system. This primary identity should hold no elevated entitlements or roles.
Another is for elevated access, such as ZTA management permissions (for creating ZT and software-defined perimeter [SDP] policies, applying patches–often automated with a PAM system, and reading logs).

4.1.2 Applications & Workloads

Imagine that the ZT implementation team tasked with determining the organization's current state has identified the IPs, applications, and workloads during the planning phase (see *Zero Trust Planning*). To satisfy access needs, these elements should be configured and organized separately at a PDP (also referred to as a controller) for onboarding, and it should be organized based on access needs. This will require centralized authentication, authorization, and monitoring, as well as segmentation of application groups.

Policies are defined at the policy administrator level to provide the high-level access needed for these assets. At the traditional stage, access needs are often organized by job function or role. This should be revamped as ZTA implementations mature through the stages; traditional, initial, advanced, and optimal, and permissions and work should instead be organized according to access needs, enabling efficient organization based on security policies.

4.1.3 Networks & Environments

To ensure the security of networks and environments from malicious actors, unauthorized visibility, and unauthorized access, organizations should embrace ZTA security best practices. These include identity-based policies, session establishment and termination, micro-segmentation, installation of PDPs (to validate authentication) and PEPs (access configuration and enforcement decisions), as well as redundant PEPs for failover and load balancing so that services can be continued even if one component fails. The ZTA diagram below summarizes one way to structure the various security components, dividing them between what is handled in the control plane versus what is handled in the data plane.

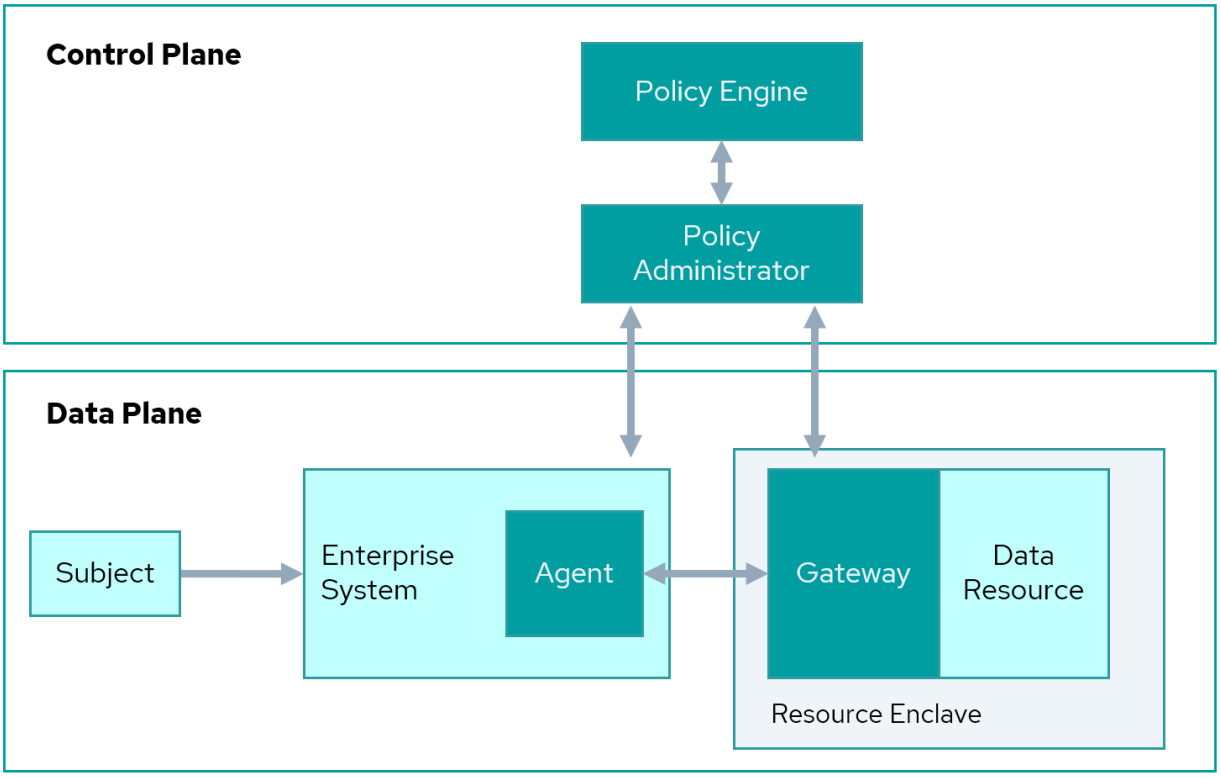

Figure 3: *Enclave Gateway Model*[12]

To assist in fulfilling the goals of this pillar, break down the network and environment considerations into these six types of network signals, which we discuss in more detail in the remainder of this section:

- Initial client authentication request reaching from agent to PEP
- Authentication request validation request (AR/VR) from PEP to PDP
- Decision transmission from PDP to PEP
- Session establishment and termination from client to resources
- Micro-segmentation
- PEP installation and access configuration

4.1.3.1 Initial Client Authentication Request

Establishing a secure connection between the application and server is essential to ensure mutual authentication. This ensures that both parties are verified before communicating with applications with sensitive data access. To initiate client authentication safely for PEP, you should:

- Position the PEP at the network perimeter while keeping other components, such as PDP and resources, on separate network segments
- Ensure that the agent can send the initial authentication request to PEP which will, in turn, forward the request to PDP
- Configure an encrypted channel for authentication request transmission

Building ZTA with user-agent-initiated access at PDP for user authentication is also possible.

[12] Figure adapted from: NIST. (2020). Zero Trust Architecture (SP 800-207).

4.1.3.2 Authentication Request/Validation Request

AR/VR is an important part of ZTA. AR/VRs help ensure that only authorized requests can be processed and approved by authorized entities, helping to prevent identity spoofing and other malicious activities.

The user agent can securely share their credentials with the PEP, which will be forwarded to the PDP for validation. The PDP then verifies the user or subject's credentials and initiates an additional MFA process. Once verification is complete, the authorization data is shared with the PEP. To ensure secure communication between the PEP and PDP, network access should be configured to allow for incoming and outgoing transmissions only between these two parties. Additionally, authentication and a secure channel of communication should be established.

4.1.3.3 Decision Transmission

Decision transmission is an essential component of ZTA, as it enables the PDP to make an informed decision about access based on user- and context-based information. This ensures that users are granted only the least amount of access required to perform their job duties, protecting data from unauthorized access.

To ensure secure transmission of data between the PDP and the PEPs, you must:

- Configure your network access to accept incoming and outgoing transmissions from only the PDP and PEPs
- Set up authentication between the two
- Perform periodic re-authentication challenges

Doing so will help to guarantee that data is kept safe and secure.

4.1.3.4 Session Establishment or Termination

To ensure secure access to their network, organizations must establish and terminate client sessions in a way that verifies the identity of clients, validates session data, and prevents person-in-the-middle attacks. Session termination is particularly important for businesses that allow privileged professionals such as company directors or medical doctors to log on from any machine within the work environment. To properly establish and terminate client sessions requesting resources, organizations should:

- Configure their PEPs to respond only to the initial authentication request
- Manage client sessions according to the authorization decided by the PDP

4.1.3.5 Micro-Segmentation

Micro-segmentation is a key component of ZTA solutions, which helps improve network security and simplify its management. Instead of creating multiple rules based on addresses, identity-based policies can be used to secure segments effectively. It is achieved by dividing resources into several

distinct network segments using either network devices, such as switches and routers, or by using host-based micro-segmentation with software agents and endpoint firewalls. The security gateway then grants access based on authorization obtained from identity attributes and must be managed to act as a PEP for protecting resources from unauthorized access.

The ultimate aim of micro-segmentation is to establish boundaries between resources within the same network zone and ensure that only authorized entities have access to secured assets.

4.1.3.6 PEP Installation & Access Configuration

Once the PEP is installed, the following checks should be conducted to improve the security stack which already supports ZT, such as port knocking and single packet authorization (SPA) for obfuscation: assess the accessibility of the device to both the PDP and endpoints at the edge of the network. The following checks can be done once the PEP is installed:

- PEPs should be able to enforce the identified authorization for access based on the policies defined at the PDP
- Ensure the PEP receives policy updates from the PDP/policy engine (PE)
- User endpoints at each in-scope location are able to reach the PEP through the network
- Ensure all in-scope PEPs and data feeds are integrated
- Establish redundant PEPs of the same kind for failover and load balancing based on scale and requirements. Similar to running a next-generation firewall (NGFW) in high availability mode, this ensures that if one PEP component fails, a second one will take over

It is important to continuously monitor the network for suspicious activity and regularly review access controls to ensure that your system's security remains intact.

4.1.4 Data

An important part of the protect surface is data (along with other resources). All data (not just information) can be better protected with a ZT implementation because ZT mandates that access decisions be made as close to the resource as possible. To apply ZT to data, it is necessary to discover, inventory, categorize (or label), and control data.

The implementations that are part of the Data pillar are done at the PDP. Using the data inventory to identify where data is located, the identity store to identify those who need access to it, and the PDP to define transaction flows for authentication and authorization of access, organizations can ensure that the necessary security measures are in place.

Logs from the PDP and PEP should be shipped to a SIEM to maintain visibility for granted accesses. This allows organizations to understand better who has access to which data and when. Ultimately, this helps them maintain the security of their data and protect their data from any potential misuse.

4.1.5 Devices

ZT can be implemented or enabled for any device in an organization. These include but are not limited to PCs, servers, mobile devices, and any OT or IoT device, to name a few. The scope for supported devices is predefined in the planning. The architecture implementation can take one of two forms:

1. Agent-based access: when a software client is installed onto the device. This sometimes encompasses other features besides the ZT agent, bundling traditional endpoint security with productivity tools or business apps; or
2. Agentless access: agentless options are deployed to devices lacking the ability to have an agent installed but can also be deployed to devices that can accept an agent. Agentless can further be subdivided into two subsets:
    - On devices that support browsers: In such scenarios, a connection using a secure tunnel to cloud/SaaS services is established through a plugin or manual configuration, which then handles all inspections.
    - On devices that cannot support browsers and agents: The entire site or micro-segment is connected using a tunnel to the cloud or SaaS service. This option may be more applicable to OT and IoT devices.

Agent-based access can also be configured on OT systems based on the implementation strategy and OT architecture.

Sometimes, you will also encounter a bring your own device (BYOD) environment, which refers to employees being allowed, and sometimes encouraged, to use personal devices to complete their work for the employer. Whether your organization's ZT policy is to use agent-based approaches or agentless access methods, BYOD scenarios will impact aspects of your ZT implementation. For example, deploying agents will require you to add a privacy notice if this agent is being installed on a personal device. Your team will need to coordinate with governance, compliance, and legal teams to confirm your messages are correct and in accordance with local law.

When an employee-organization agreement terminates, you need to ensure that the decommissioning of ZT-related agents and apps involves the decoupling of security solutions from a BYOD device. To complete these tasks, employees need to know the policy and procedure. They may also need to be reminded that some of the protection that they previously relied upon is being removed.

4.1.5.1 Deploying Agent-Based Access

In the case of agent-based approaches, a software agent must be installed and run on all endpoints. It is then the agent's job to collect the user identity and share the security posture data of the device and connection. Agents must be regularly updated – either automatically or as part of the company's patch process. To ensure compatibility with different device types, such as Mac OS, Linux, and Windows laptops, they should be tested appropriately before being deployed on end-user devices. The setup process and usability factor should also be evaluated prior to mass deployment.
The available options will depend heavily on the target environment (e.g., Windows, Linux, Android,

iOS), and the selected solution may vary in how the agent is finally deployed.

Based on current IT trends, agent deployment will likely be based on unified endpoint management (UEM) and mobile device management (MDM). However, most vendors provide a download console for agent installs.

4.1.5.2 Deploying Agentless Access

The agentless access method can be used to deploy ZT onto an endpoint device, such as with a browser or browser-based application. The browser is responsible for assessing the security posture and connection of the device prior to allowing access. This deployment method is also useful when dealing with light devices that do not have a browser available, like OT devices. In this case, a verified network with a proxy handling the agent load and building the connection can be established. After authentication of the user through an IdP (which may incorporate single sign-on [SSO] or MFA), they will then be redirected to their requested resources if authorization is granted.

4.1.6 Visibility & Analytics

Implementing ZT requires a platform-based approach to security, empowered with analytic and visibility dashboards that authorized personnel can use to make policy changes. To achieve this, agents or APIs should be implemented to gather logs from various log sources.

Depending on the requirements of the log aggregation tool, a log collector may be necessary. Aggregated logs enable event correlation that is more powerful in threat analysis and discovery than if the logs are kept separate and analyzed separately. The visualization component of the central log aggregation tool can provide dashboards, enabling visibility of the inner workings of each pillar.

To achieve the above-mentioned set of goals:

- Implement agents or APIs to acquire logs from each of the logs' sources.
- If required, implement a log collector.
- Ship the logs to the log aggregation tool using either a push or a pull mechanism, depending on the tool's requirement.
- Implement a search tool that can provide a query language that can be used to search logs for events and event correlation.
- Design dashboards to view query and event correlation output.

Once a visibility dashboard has been fully implemented, potential threats can be identified and monitored through careful observation of the dashboards.

4.1.7 Automation & Orchestration

The ability to orchestrate and automate the deployment of the target architecture's logical components is key to a successful ZTA implementation. This includes tasks such as updating the ZTA components' security posture, dynamic access and authorization policy updates, patch

management, and change management. These deployments can span on-premises, cloud, or hybrid implementations.

Depending on the deployment model identified in the target architecture, two types of orchestration methods can be used for ZTA deployments: application pipeline and infrastructure as code (IaC) pipeline. In most cases, a combination of both is used. The IaC pipeline is typically used when the PEP is a single component acting as a gateway for the subject requests. In contrast, the application and pipeline are commonly used when an AuthZ module and other modules must be deployed directly on the component. This could include deployment within the application code.

Depending on the orchestration methods and target architecture, different functional and non-functional orchestration requirements will exist. However, it is important to note automation and orchestration can be achieved in some cases but not all. For example, achieving it in OT or industrial control systems (ICS) is very challenging. It is important to factor in this while implementing ZT for such technologies.

4.1.8 Governance

Governance of the ZT program and individual ZT projects is essential to ensure successful implementation and control over goals, requirements, and actions taken. A formal procedure for governance should be established through a review committee that will evaluate the progress made towards meeting objectives, ensuring that plans are funded, and assessing associated risks with future phases.

As part of a successful ZTA implementation, it is essential to establish a formal review process headed by senior management. This committee will ensure that appropriate ZT requirements are observed and that the organization has the necessary resources to complete the ZTA plan. Their main objectives include:

- Verifying that each phase is completed with success
- Ensuring sufficient funding for the next phase
- Assessing the risks associated with continuing to the following phase[13]

At the start of implementation, metrics need to be defined and collected to measure set parameters. These metrics can range from high-level indicators, such as the number of goals achieved, budget consumption, and impact on organizational policies, to lower-level metrics, which include the number of support tickets raised, complaints by end-users, policy changes, failed policies, and downtime incurred. All these should be identified.

4.1.8.1 ZT Policies

ZT policies are used to bridge the gap between a business's mission and its risk management requirements. These policies are documented and set up within the ZT planning process. The PDP and PEP communicate in near real-time to ensure that authorization decisions made by the PDP,

[13] United States Department of Health and Human Services. (2012). Enterprise Performance Life Cycle Framework.

based on the policies defined at the PE, are enforced. Policies can be applied to users, applications, and workloads during the PDP onboarding stage. These policies will define access conditions for each user or device based on parameters such as location and time.

To effectively manage a ZTA implementation, it is important to save some of your authorization work for the implementation stage. This is because a macro-level approach will focus too much on the overall architecture and will create extra work each time the overall architecture changes. Instead, your detailed policy rules should focus on each PDP and PEP technology separately. For example, a privileged access workstation (PAW) can only be accessed by a specific user or administrator from their approved device. In that case, there are various places where the policy could be updated. One could be the local permissions on the PAW host. Another could be the micro-segmentation firewall rules for the user and device hostname. Lastly, a network access control (NAC) or UEM solution could check the user's local device posture.

> - Attack surfaces change quickly, altering risk. Policies must be updated regularly by:
> - Re-evaluating the updated transaction flows for changes in risk
> - Re-implementing a macro- and micro-level approach to access and authorization policies
> - Ensuring that updated policies are applied to the policy engine at the PDP

4.2 Transaction Flow Architecture Review

Maintaining a transaction inventory allows you to reevaluate the behavior of the data within each transaction at regular intervals and, more importantly, detect any changes or abnormalities.

In the planning phase (see *Zero Trust Planning*), we discussed the need for a detailed analysis and mapping of existing transaction flows. When the time comes to implement any identified changes to these transaction flows as part of your ZTA implementation, there will be some considerations to address, such as the addition of ZT nodes, services, and components. You will also need to address legacy controls that need to be replaced to protect existing transaction flows.

Mappings may need to be reviewed against your planning notes. This will happen often, both during and after your implementation. As we discussed in the previous section (ZT Policies), you don't want to focus too much on the overall architecture; not only will it be too complex to manage easily, but it will create extra work each time any part of the architecture changes. Instead, your detailed transaction flows should focus on each individual protect surface.

By orienting your transaction flow diagrams or transaction inventory to each protect surface, you can easily manage change down the road.

4.2.1 Transaction Flow Mapping

In the context of ZT, we could record the transactions that are part of the subject-initiated access to the applications and network or between the access-controlled applications, as long as we are centered on the surface we need to protect.

The way that you map your flow is always unique to your implementation. More often than not, this process is manual. While some automation tools to help you with mapping are emerging, such technology is still catching up to the need.

The best way to maintain a transaction inventory of the transactions you are managing, along with their dependencies, will depend on the unique characteristics of your team and the organization's IT architecture. Nevertheless, in the next section, we provide you with a brief example.

4.2.2 Converting Flow Maps to Transaction Lists

In the example below, the management team chose to invest the time to map all transactions involving their protect surfaces with a professional flowchart tool that their network engineers are proficient in. This will help them communicate with the vendors they use to receive security-oriented services. However, they also want to have each transaction listed because they plan on using this as a checklist during implementation and, down the road, as a potential set of requirements that can guide future development projects.

The ZT implementation team determined that this particular protect surface involves six transactions, each with a unique identifier (see Figure below). Dependencies are shown.

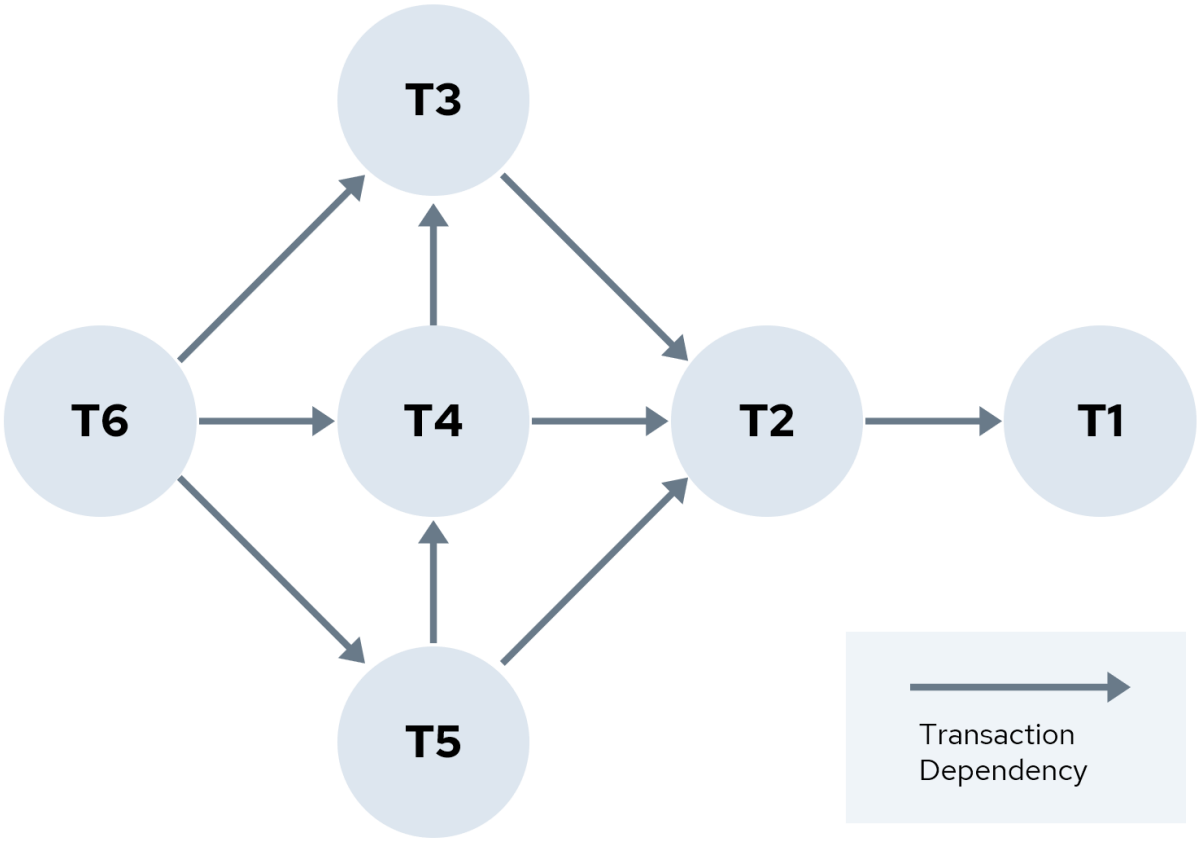

Figure 5: *Transaction Inventory*

Next, each transaction path is assigned a **Transaction ID**. Any transaction it depends upon is entered in **Input**, and any transactions dependent upon that identified transaction are labeled **Output** (see Table below). The table can also include optional values, such as a **Service ID**, a **Title** (not shown), or **Description** (not shown).

Transaction ID	Input	Output	Service ID
Transaction 1 (T1)	T2		Service Baseline A
Transaction 2 (T2)	T3, T4, T5	T1	Service Baseline B
Transaction 3 (T3)	T6, T4	T2	Service Baseline C
Transaction 4 (T4)	T6, T5	T3, T2	Service Baseline D
Transaction 5 (T5)	T6	T4, T2	Service Baseline E
Transaction 6 (T6)		T3, T4, T5	Service Baseline F

Table 2: *Transaction Configuration Management Inventory*

This table can be entered into a configuration management system, unified modeling language (UML) tool, or programmable logic controller in reverse. Such tools can help provide indexable and searchable data that can help troubleshoot problems and document fixes. You can even use it to create more complex ladder or signaling diagrams.

4.3 Testing

After an organization has completed implementation, it must develop and maintain relevant policies, procedures, and agile testing scripts that define how the ZT testing process works and should be conducted so that the testing methodology remains consistent from one implementation stage to the next. To ensure that the planned ZTA delivers the intended service levels, and before the legacy architecture can be decommissioned, a test cycle must be completed for ZT implementation, including testing on non-production and production environments. It is important to isolate whether a problem originates from the implementation itself or is the result of new technology merely catching a problem that existed in a pre-existing solution or data but was not caught due to weaknesses in the previous setup.

Security testing must ensure that access controls are working correctly and the network is protected from threats. This testing includes vulnerability scans, penetration tests, application security assessments, system readiness testing, operational readiness testing, and other forms of security testing. Testing must:

- Confirm that the ZT objectives were achieved.
- Provide evidence that continuous authentication and authorization over all communications, users, systems, and networks is taking place.

- Provide evidence that employees are able to complete their work with a minimum of disruption.
- Create a secure baseline from which future changes can be monitored for potential threats or vulnerabilities.
- Confirm that there is a robust audit trail to monitor suspicious activity and policy violations.
- Ensure that the organization is on the path to maintaining effective communication between monitoring systems and personnel to ensure the efficiency and thoroughness of the ZT process.

Regardless of the test type or environment you are testing, each testing activity must align or refer back to the ZT plan and strategy. Integrate ZT testing efforts into the overall cybersecurity program; doing so will provide enhanced protection. Additionally, each planning objective in each pillar needs to be sufficiently tested to confirm that your team has met its objectives.

Finally, promote collaboration between departments to ensure ZT testing results in secure and efficient operations across the organization. When all test phases are completed, production can begin cut-over, also in phases.

4.4 Continual Improvement

During implementation, the ZT project should proceed in a continuous feedback loop, with each pillar, sub-project, and related effort recorded in a task repository that can be analyzed by a project management expert and analysis tool.

However, don't stop with task monitoring. Additionally, organizations should perform regular audits to ensure their policies and practices are followed. These audits should test current security measures and identify any potential gaps in the system that malicious actors could exploit. This way, organizations can ensure their networks are secure and compliant with best practices.

This feedback loop drives future ZT adjustments, as needed, in response to any encountered challenges, timeline modifications, or other reasons. ZT projects can take considerable time to implement. Any technology changes in the environment require review and consideration at the ZT project level to achieve the original goals and possibly make adjustments to those goals, with approval from all relevant stakeholders.

A key component of any ZT project should be proper risk management. As the ZT project is implemented and adjustments are made upon encountering a changing environment, the feedback loop should trigger a re-evaluation of risks to the ZT project, which, in turn, triggers ZT-related change control.

4.5 Project Closure

Successful ZT implementation would include a complete inventory of all transactions, dependencies, and services with associated IDs. Policies and procedures should be developed to ensure consistent testing methodology across different stages of the implementation. Security tests such as

vulnerability scans, penetration tests, application security assessments, and system readiness testing should have been conducted to protect the network from potential threats. Tests should be carried out on both non-production and production environments to ensure the quality of the implementation. All legacy architecture should be decommissioned once all tests have been completed successfully. Finally, sufficient documentation must be created to ensure future troubleshooting is easier and fixes can be documented accurately.

With all the project-related tasks completed, the project needs to be formally closed. For successful operations and maintenance, important policies, procedures, and processes must be reviewed and maintained. Final sign-offs must be obtained from key stakeholders, and a "go-live" date must be communicated to end users. From here, operations and maintenance cycles will then follow.

Conclusion

ZT is an important security strategy that must be planned, tested, and monitored for optimal effectiveness. ZT implementations must be iterative so that their efficacy can be evidenced while assessing results; this better prepares future implementations that lead to higher ZT performance levels with the ultimate goal of having proactive monitoring and logging feedback before a malicious actor can achieve a breach.

Furthermore, organizations should integrate ZT testing into their overall cybersecurity program to provide enhanced protection. Additionally, regular audits of policies and practices should be conducted to identify potential security gaps in the system. A key component of any ZT project should also involve proper risk management and knowledge management to ensure lessons learned from incidents are not repeated.

With the help of these strategies, organizations can increase their ZT efficiency and proactively protect their operations or security against malicious actors.

Glossary

Please refer to our Cloud Security Glossary, a comprehensive glossary that combines all the glossaries created by CSA Working Groups and research contributors into one place.

Acronym List

Acronym	Term
API	Application programming interface
AR/VR	Authentication request/validation request
BYOD	Bring your own device
BCP	Business continuity planning
CSA	Cloud Security Alliance
CISA	Cybersecurity and Infrastructure Security Agency
DR	Disaster recovery
DNS	Domain name system
IdP	Identity provider
ICS	Industrial control systems
IaC	Infrastructure as code
IoT	Internet of Things
IP	Internet Protocol
IDS	Intrusion detection systems
IPS	Intrusion prevention systems
MDM	Mobile device management
MFA	Multi-factor authentication
NAC	Network access control
NAT	Network address translation
NGFW	Next-generation firewall
OSI	Open Systems Interconnection
ORT	Operational readiness testing
OT	Operational technology
PC	Personal computer
PDP	Policy decision point
PEP	Policy enforcement point

PE	Policy engine
PIP	Policy information point
PAM	Privileged access management
PAW	Privileged access workstation
SIEM	Security information and event management
SOAR	Security orchestration, automation, and response
SMS	Short message service
SPA	Single packet authorization
SSO	Single sign-on
SaaS	Software as a service
SDP	Software-defined perimeter
SRT	Systems readiness testing
UEM	Unified endpoint management
UML	Unified modeling language
ZT	Zero Trust
ZTA	Zero Trust Architecture
ZTT	Zero Trust Training