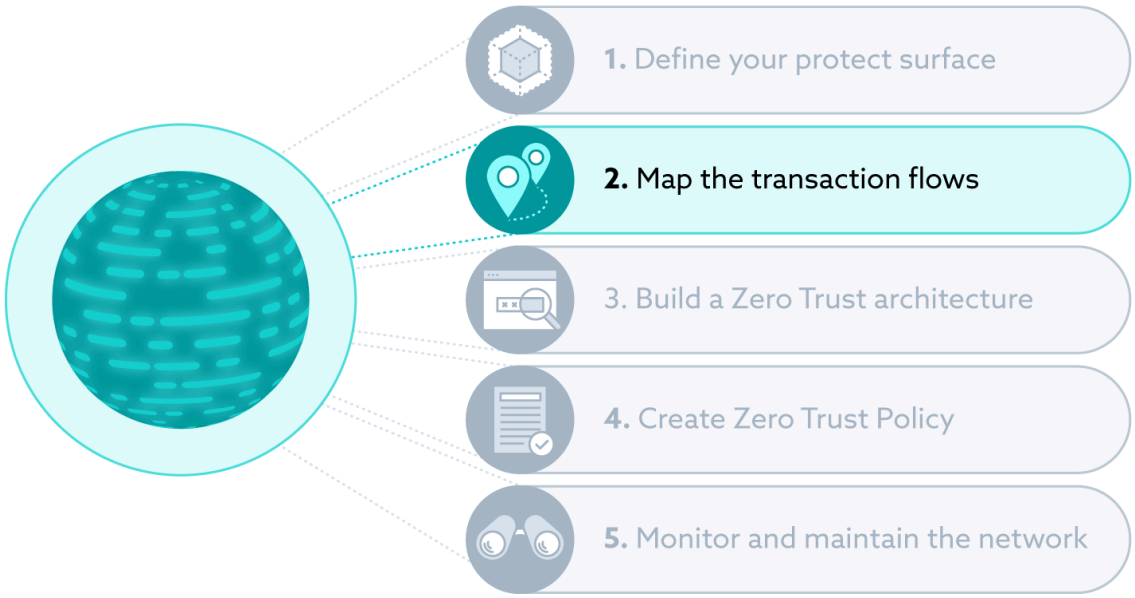
cloud security alliance®

Map the Transaction Flows for Zero Trust

Step Two of the Five Step Process



Acknowledgments

The CSA Zero Trust Research Working Group

The scope of Zero Trust research and guidance necessarily includes cloud and on-premises environments along with mobile endpoints and applies to the Internet of Things (IoT) and operational technology (OT). The goals of the CSA Zero Trust (ZT) Working Group are to:

- Collaboratively develop and raise awareness of Zero Trust best practices as a modern, necessary, and cloud-appropriate approach to Information Security (InfoSec).
- Provide thought leadership and educate the industry about the strengths and weaknesses of different ZT approaches so organizations can make informed decisions based on their specific needs and priorities.
- Take a deliberate product- and vendor-neutral approach to architectures and implementation approaches for mature Zero Trust implementations.
- Take technically sound positions on Zero Trust and make defensible recommendations while remaining product- and vendor-neutral.
- The workstream(s) for this document is ZT5 - Pillar: Networks, led by Vinotth Ramalingam.

Lead Authors
Vinotth Ramalingam
Michael Roza

Contributors
Rajesh Murthy
Steven Guilford
Jackson Daniel
Vaibhav Malik

Reviewers
RamaKrishna Manchana
Taresh Mehra
David Nance
Jerry Chapman
Angell Duran
Jason Garbis
Chris Jablonski
Erik Johnson
Shruti Kulkarni
Chandra Rajagopalan
Sean Connelly
Justin Bowen
Govindaraj Palanisamy

CSA Global Staff
Erik Johnson

Table of Contents

Abstract

Zero Trust has emerged as a paradigm shift in cybersecurity strategy, advocating a "never trust, always verify" approach. The objective of this document is to provide guidance for iteratively executing the second step in the five-step Zero Trust implementation process described in the NSTAC Report to the President on Zero Trust and Trusted Identity Management, originally formulated and socialized by John Kindervag. Separate CSA research documents are being developed to elaborate detailed guidance for each of the five steps, which starts with step 1: *Defining the Zero Trust Protect Surface*.

A key step to successfully implementing Zero Trust is mapping transaction flows for the Protect Surface to understand how the business system being protected works, with the goal of defining and enforcing security policies in subsequent steps. This paper provides guidance on mapping transaction flows for the Devices, Assets, Applications, and Services (DAAS) elements comprising the Protect Surface, establishing granular visibility into communication between these elements, with other Protect Surfaces, and with users and external services.

As data flows are mapped and analyzed in detail, Protect Surface metadata and documentation are assembled, validated and refined to provide necessary inputs and understanding for building the Zero Trust architecture and defining appropriate Zero Trust policies. The five step Zero Trust implementation process is incremental, progressively elaborated and iterative.

Target Audience

- **Primary**: Zero Trust implementation teams, Information Security Managers/Officers, Security Architects, Security Analysts/Operators, Network Architects/Designers, Network Security Managers, IT Infrastructure Managers, Business System Owners
- **Secondary**: Chief Information Security Officers (CISO), Threat Modelers, Incident Managers, Auditors, Business Process Owners, Compliance Officers, Risk Managers, Network Administrators, Cybersecurity Analysts, IT Compliance Analysts

Introduction to Zero Trust

The National Security Telecommunications Advisory Committee (NSTAC) Report to the President on Zero Trust and Trusted Identity Management defines Zero Trust (ZT) as *"a cybersecurity strategy premised on the idea that no user or asset is to be implicitly trusted. It assumes that a breach has already occurred or will occur. Therefore, an entity should not be granted access to sensitive information by a single verification done at the enterprise perimeter. Instead, each user, device, application, and transaction must be continually verified."*

Traditional, centralized, trust-based "castle and moat" physical network perimeter security architectures are ineffective in the current era of decentralized cloud computing and remote workforce, where few organizational assets and users still reside inside the "castle."

Sophisticated threat actors are increasingly adept at exploiting any exposed technical or human vulnerability in modern, highly distributed enterprise networks that often leverage Internet connectivity heavily. Successful cyberattacks exploit trust somehow, making "trust" a dangerous vulnerability that should be mitigated and managed.

Zero Trust is a holistic enterprise security strategy that encompasses cloud/multi-cloud (all service models), on-premise/hybrid systems, internal and external partner/stakeholder user (organization managed and BYOD) endpoints, and is inclusive of operational technology (OT), Industrial Control Systems (ICS) and Internet of Things (IoT). Consequently, Zero Trust has been compared to a mountain that must be climbed one step at a time, that is, implemented incrementally and preferably in a risk-based manner. These principles are a common theme in CSA ZT guidance.

Enterprise adoption of Zero Trust is broad and growing. VentureBeat reports that 90% of organizations moving to the cloud are adopting a Zero Trust Strategy,[1] while Gartner predicts that 10% of large enterprises will have mature Zero Trust programs in place by 2026.[2]

Document Scope

The objective of this document is to provide detailed guidance for iteratively executing the second step in the iterative five-step Zero Trust (ZT) implementation process described in the NSTAC Report to the President on Zero Trust and Trusted Identity Management (pg. 7), which was initially developed and socialized by John Kindervag, an originator of Zero Trust.

This paper focuses on executing Step 2, Mapping the Transaction Flows between the Data, Applications, Assets, and Services (DAAS) elements comprising the business system that constitutes the Protect Surface in question for the current iteration of the five step process.

[1] https://venturebeat.com/security/why-90-of-enterprises-migrating-to-the-cloud-are-adopting-zero-trust/
[2] https://www.gartner.com/en/newsroom/press-releases/2023-01-23-gartner-predicts-10-percent-of-large-enterprises-will-have-a-mature-and-measurable-zero-trust-program-in-place-by-2026

This mapping includes data flows with and between users, devices, supporting services (e.g., IDaaS, SIEM, SASE), other Protect Surfaces/business systems within the organization, as well as external systems/services operated by other organizations. It explores various methods for mapping these flows, including comprehensive system analysis and leveraging scanning tools. The document also delves into the role of transaction flow mapping in refining Protect Surfaces and designing Zero Trust architectures. Additionally, it outlines the benefits of mapping transaction flows and provides a maturity model for evaluating the effectiveness of transaction flow mapping practices.

The document concludes with significant insights into Step 3, “Building a Zero Trust Architecture,” which focuses on designing the Zero Trust architecture tailored to the Protect Surface(s) defined in Step 1, whose transaction flows have been mapped in Step 2.

Zero Trust Implementation Process

This document provides guidance for executing the second step defined in the 5-step Zero Trust implementation process, as described in the NSTAC Report to the (US) President on Zero Trust and Trusted Identity Management and further elaborated in other Useful References. This foundational reference document, which CSA Zero Trust research leverages and aligns with, describes the five-step method as an iteratively executed repeatable process in section 2.1.1, which begins with developing an inventory of the organization’s business systems (protect surfaces) for risk-based prioritization for Zero Trust protection based on their criticality (highest first) and then their current level of security maturity (lowest first).

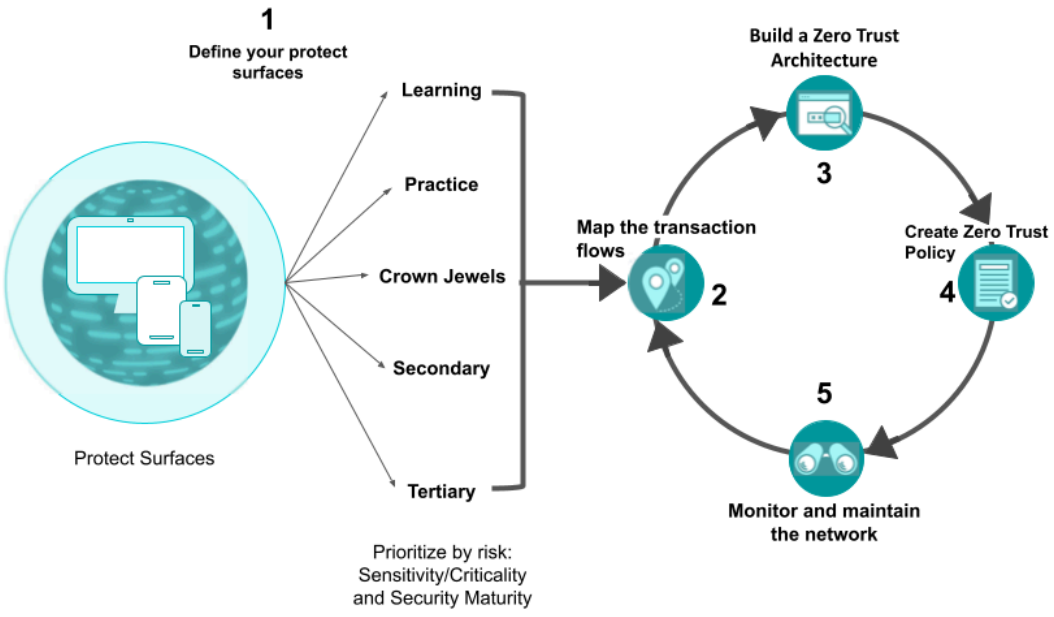


Figure 1: Iterative Five-Step Process for Enterprise Zero Trust Implementation

The five-step process for iterative ZT implementation can be summarized as follows. The CSA is developing detailed guidance for each of the steps, along with application guidance for specific use cases such as critical infrastructure and small and medium sized businesses (SMBs).

1. Defining the Zero Trust Protect Surface - identifying business system assets, determining their business criticality and risk classification, and assessing their current security maturity to prioritize their ZT security implementation over a series of process iterations.
2. Mapping the transaction flows involves thoroughly analyzing and documenting the data flows and security maturity for the Protect Surface's DAAS elements to inform Zero Trust architecture design and policy creation.
3. Building a Zero Trust architecture encompasses implementing security across all five pillars with cross-cutting capabilities, emphasizing strong authentication, network segmentation, data encryption and comprehensive logging and monitoring.
4. Creating Zero Trust Policy involves defining and enforcing granular access policies for the protect surface as close as possible to the business assets being protected, along with incident detection and response mechanisms.
5. Monitoring and Maintaining the network requires continuous vigilance, including analyzing logs, scanning for malware, detecting and responding to incidents, and conducting thorough security testing to identify and address vulnerabilities promptly.

Overview of Mapping Transaction Flows

Successful Zero Trust security implementation for an organization relies on understanding its business systems, including their users, components, functions, and interfaces. It's a progressively elaborated process that begins with identifying the Protect Surfaces, their constituent DAAS elements and ZT implementation priority in Step 1. In Step 2 the initial Protect Surface metadata is validated and refined by understanding each DAAS element, transaction and interaction to define and implement the right granular ZT security policies. Organizations can start with critical business systems and gradually expand them to all other systems.

Depending on the organization's business environment and requirements, a Business Information System constitutes one or more Protect Surfaces containing DAAS elements. CSA's guidance on Defining the Zero Trust Protect Surface discusses the first step to identify the DAAS elements of a Protect Surface. At the same time, this paper elaborates on the next step to understand user and business process interaction with each Protect Surface: analyze, comprehend, and document the transactions associated with each component of the defined Protect Surface and refine the Protect Surface based on its understanding. It facilitates a thorough understanding of business information systems, enabling the placement of appropriate security controls for each system and its subsystems.

The fundamental objective of Step 2 is to understand and document the operation and current security maturity of the business system being protected to facilitate the designing and

implementation of a cost-effective ZT security architecture and incremental, risk-based approach to enhancing and maturing its security posture over time.

Step 2 activities include understanding and documenting the following:

- Resources that access applications, data, and infrastructure, such as users, APIs, web services, etc.
- The relationship between each application and its data, including but not limited to acquisition, transfer, movement, and persistence
- How DAAS elements within the Protect Surface interact and communicate with each other and with resources outside the Protect Surface
- How traffic involving the DAAS elements traverses the network and flows across different segments or boundaries
- The business context and rationale behind each interaction and data flow

By mapping and documenting these operational aspects of the system, organizations can gain valuable insights into:

- How to effectively segment the network around each Protect Surface
- Where to strategically place security controls and enforcement points
- Which traffic patterns and communication channels need to be explicitly authorized or denied based on policies
- How to design a ZT architecture that aligns with the specific data flows and the functionality and interactions of the DAAS elements comprising the Protect Surface
- Elements of identities, devices, and networks will surface in the access policies and eventually be created for each Protect Surface

We leverage the concepts from the Defining the Zero Trust Protect Surface paper and build on them with the Step 2 activities described below to map transaction flows to help understand how the system works and where the security weaknesses are.

Step 1: Activities to Define your Protect Surface(s)

- ➔ Inventory the organization's Business Systems/Protect Surfaces and their constituent DAAS elements
- ➔ Assess and document the business sensitivity/risk classification and security maturity for each system
- ➔ Prioritize most sensitive systems for ZT implementation, starting with those with the lowest security maturity (usually after a couple of practice runs)

The output from Step 1 is the inventory of Business Systems/Protect Surfaces with an initial set of key metadata for each.

Step 2: Activities for Mapping the Transaction Flows

- ➔ Validate the Protect Surface's DAAS elements and refine their metadata
- ➔ Identify and document users of the Protect Surface and the types of endpoint devices they use
- ➔ Identify dependencies, such as related business systems and external services
- ➔ Identify transactions among DAAS elements, with users, other business systems and external services

- ➔ Map the data flows and document how the system works based on this information
- ➔ Validate and refine the understanding of the relative sensitivity and security maturity of the system and its DAAS elements

How to Map Transaction Flows

Now that we have described the activities involved in mapping transaction flows and understanding business system operations, we can see that gathering this detailed information can be a complex task in the digital landscape where data may traverse multiple platforms, service providers, geographic regions, and user populations. For Zero Trust, the set of transaction flows is intended to include all information and data flows, not just financial transactions or particular types of sensitive information. Organizations should understand all data flows and the functions of every constituent DAAS element, which can span different environments and organizations. This requires a deep understanding of business system operations and the functions of and interfaces to its DAAS elements.

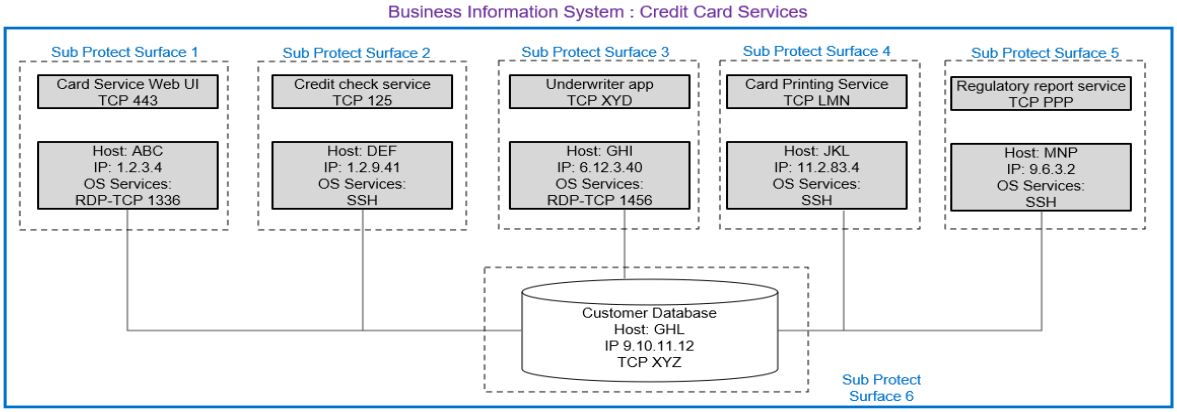
Example: Here are some of the best approaches explained with an illustrative example using a fictitious financial services organization that offers consumers credit cards, mortgages, deposits, and insurance services as an example to elaborate on the steps for mapping transaction flows.

Step 1: Defining the Zero Trust Protect Surface. This step identifies all of an organization's Protect Surfaces, along with their DAAS elements, and assesses their business criticality and current security maturity for prioritization purposes. For this discussion, let's pick a *Credit Card Service* as our critical business system and map its transaction flows.

The Credit Card service has a number of integral functional DAAS elements and interfaces, including:

- **Credit Card Web UI**: This is used by customers and sales representatives applying for a credit card to perform online self-services for managing the account.
- **Credit Check Transaction Service**: Once an application is submitted, this service communicates with external credit bureaus to verify and report the customer's credit history.
- **Underwriter Application**: A back-end application designed for underwriters to process credit application requests, incorporating inference or scoring algorithms for auto-approval based on credit information and facilitating manual reviews by underwriters when necessary.
- **Card Issuing Application**: Upon approval, this application generates a unique card number and manages associated discretionary data necessary for account management.
- **Card Printing Service**: The generated card details are transmitted to the card production environment or external service providers to manufacture physical cards and distribute them to customers.
- **Regulatory Report Service**: Generates and sends monthly reports to Federal agencies in compliance with regulatory guidelines.

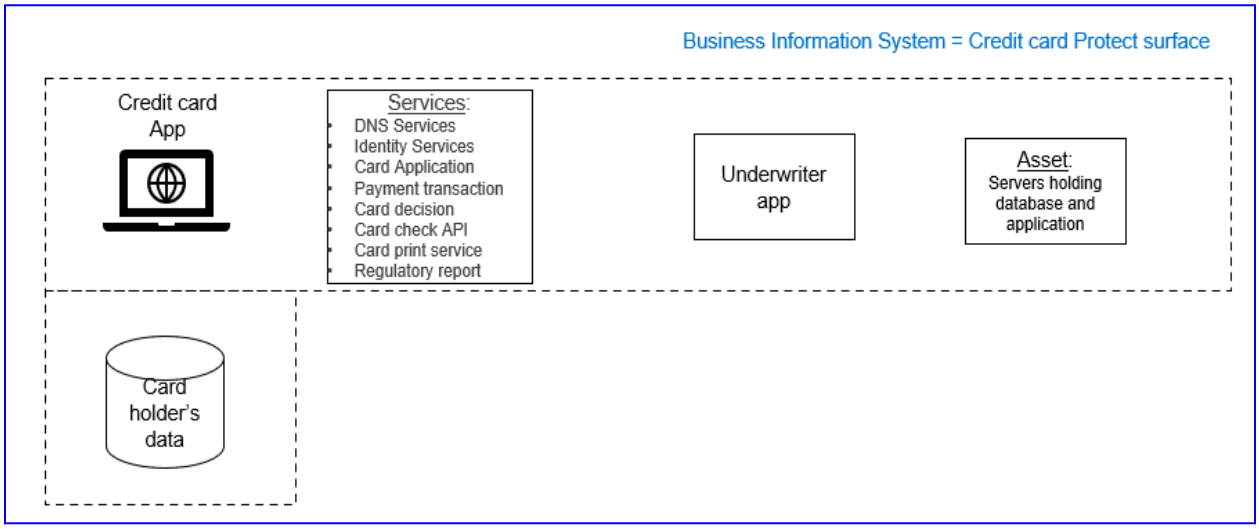
Each of these can be considered subsystems of the Credit Card business system. This detailed understanding will enhance the analysis of functionality, data flows and interactions, which will, in turn, aid in defining tailored ZT controls for each subsystem and DAAS element.



Validate Protect Surface DAAS Elements

Engage with relevant teams (e.g., development, solution architects, operations, security) to validate the completeness and operational role of the set of the system's DAAS elements.

- **Data**: *Card holder's data,* Personally Identifiable Information (PII)
- **Applications**: *Credit card web application, Credit Underwriter application*
- **Assets**: *Servers hosting applications and database*
- **Services**: Card application, Payment transaction, Card decision, Card check, Card print and report generation services, DNS services, Identity services.



Identify Business System Users

Identifying all human users of a Protect Surface involves understanding all internal and external stakeholders. This includes internal users, Business-to-Customers (B2C) and Business-to-Business (B2B), third-party service providers, suppliers and contractors, guest

users, etc. Here, we have summarized a list of examples of different user types with systems and tools to collect that information.
Similarly, all non-human users and their functions and roles need to be identified.

Human User Type	List/Examples	Source
Internal users	Employees, contractors, temporary workers, system administrators, and other privileged users	Identity and access management (IAM) systems, employee directories, HR records, and access logs
External users	Customers, partners, suppliers, vendors, third-party service providers, guest users and the general public (B2C) may include some privileged users (e.g., for delegated administration)	Customer databases, partner portals, vendor management systems, and collaboration platforms (identify external user accounts and profiles)

Credit Card Business System Example

- Identify human users of Protect Surface - Customers, Credit sales reps, and Credit underwriters.
- Identify non-human users, including APIs and external interface identities. These are typically associated with system interdependencies, interactions, and interfaces, as described in the next section.

Identify Dependencies and Interactions

Identifying dependencies of a Protect Surface involves a thorough analysis of system operations, processes, data flows, and relationships that support or interact with the business system's DAAS elements. This includes identifying all non-human identities within an organization or external that have access to and interact with Protect Surface.

Internal dependencies are dependencies with other Protect Surfaces. These include:

- Dependencies with other applications or protect surfaces that exchange data for business functions
- Dependency on specific software libraries, frameworks, APIs, or middleware used by applications

Process dependencies are dependencies related to business processes, workflows, and operational procedures within the Protect Surface, including:

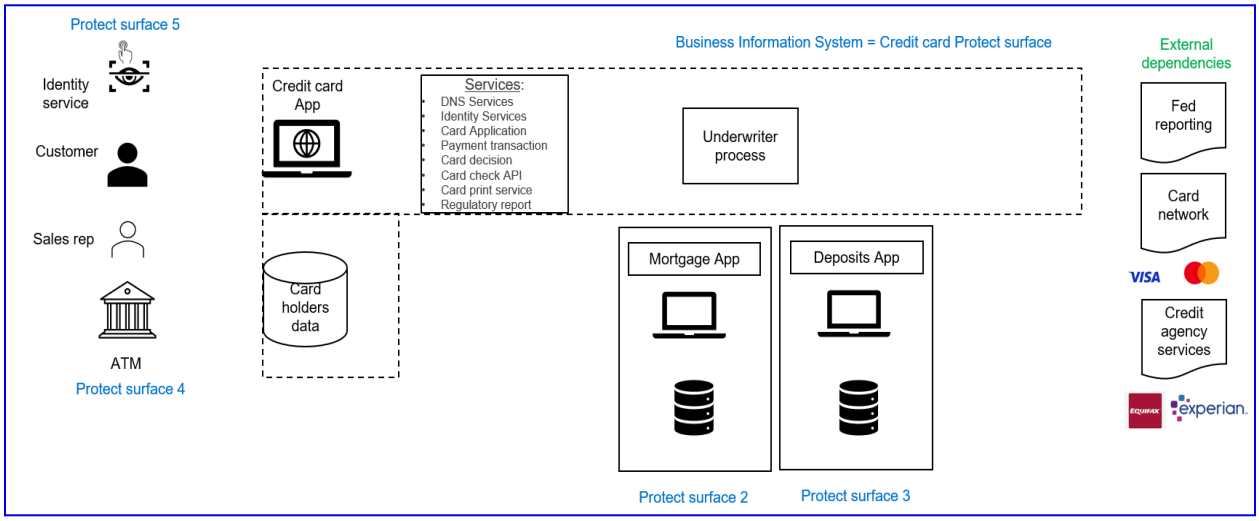
- Dependencies on specific workflows, tasks, or operations that rely on interconnected systems, applications, or resources
- Dependencies on manual or automated processes, including scheduling, batch processing, job dependencies, and workflow triggers

External dependencies are dependencies that interact with, or support, the Protect Surface from outside, such as:

- Third-party services, APIs, cloud platforms, and SaaS applications used by the applications
- External data sources, feeds, or APIs that provide input or contribute to data processing within the Protect Surface
- Dependencies on external vendors, suppliers, partners, and service providers needed for business operations
- Dependencies related to regulatory compliance, industry standards, contractual obligations, or external policies that impact the Protect Surface

Credit Card Business Information System Example

- Identify dependencies
  - External services: *Credit agency, Fed reporting, ATM*
  - Other Protect Surfaces: *Active directory, Mortgage services, Deposit services, Email and messaging services, Identity services, Logging and forensic services, IPS and IDS*



Understand How the System Operates

After identifying the users and dependencies of the business system, the next step is to understand and document the functions and interactions of DAAS elements, which involves understanding how data is accessed, processed, transmitted, and shared across various components and stakeholders. While there could be many ways to achieve this, below are the

two best approaches that we believe can be employed to understand functions, interactions and transaction flows.

1. **Comprehensive System Analysis - Architecture, Data flow, and User interaction analysis**: Use available resources such as network diagrams, application architecture diagrams, threat models, user interaction diagrams, and data flow diagrams to gain insights into the existing business information system.

2. **Leverage Scanning and Monitoring Tools**: Scanning tools such as firewalls, extended detection and response (XDR), and log monitoring tools are used to view and analyze traffic and develop comprehensive transaction flow mapping.

Comprehensive Systems Analysis - Architecture, Data Flows & Users

Effective transaction flow mapping for a Zero Trust Protect Surface can be created by leveraging existing or creating new, up-to-date artifacts for the business system, such as architecture diagrams, network diagrams, user interaction diagrams, and data flow diagrams. This in-depth understanding is gained through engaging with the business system owners and various stakeholders across the network, application, enterprise, and business teams to understand the system, how it works, and the key risks.

User Interaction Diagrams: Review user interaction diagrams or flow charts illustrating how users interact with the system, including user actions, input/output data, and user interface components. Understanding user interactions is crucial for mapping end-to-end transaction flows.

Stakeholder Collaboration: Engage with stakeholders, including business system owners, network architects, application architects, Enterprise architects, data stewards and custodians, and related business area representatives, to gain a deeper understanding of system requirements, functional specifications, and operational constraints and learn the purpose of applications, business information flows, and the business processes they support.

Analyze and validate existing artifacts:

- **Network Diagrams**: Gather network diagrams that visualize the network topology, highlighting physical and logical connections between various DAAS elements like servers and network devices.

- **Application Architecture Documentation**: Reviewing application architecture diagrams provides a comprehensive understanding of the software system structure. This includes placing assets, resources, data, and services and how critical components like databases, APIs, microservices, and user interfaces interact during a transaction flow. Over the past few years, application architectures have transformed from monolithic designs to more modular and distributed approaches such as microservices, containers, and serverless computing, which organizations must consider while analyzing.

Modern applications often employ multi-tier architectures (N-tier architecture), where the functionality is distributed across multiple layers, such as the presentation, business logic, and data tiers. Mapping transaction flows in such architectures requires a thorough understanding of how data is accessed and manipulated within each tier and how they interact. When mapping transaction flows in a multi-tier architecture, it is important to consider the following aspects:

- **Data flow between tiers**: Understand how data is passed between the presentation, business logic, and data tiers. Identify the protocols, architecture styles, and formats used for communication, such as HTTP, REST, or SOAP.
- **Data transformation and validation**: Analyze how data is transformed, validated, and processed within each tier. Identify any data manipulation, formatting, or enrichment at each layer.
- **Access controls and authentication**: Assess each tier's access controls and authentication mechanisms. Ensure that proper authentication and authorization checks are in place to prevent unauthorized access to data.
- **Error handling and exception management**: Examine how errors and exceptions are handled within each tier and how they are propagated across the tiers. Identify any potential security vulnerabilities that may arise from improper error handling.
- **Caching and data storage**: Consider using caching mechanisms and data storage at each tier. Understand how data is cached, stored, and retrieved within the application and assess the security implications of these practices.

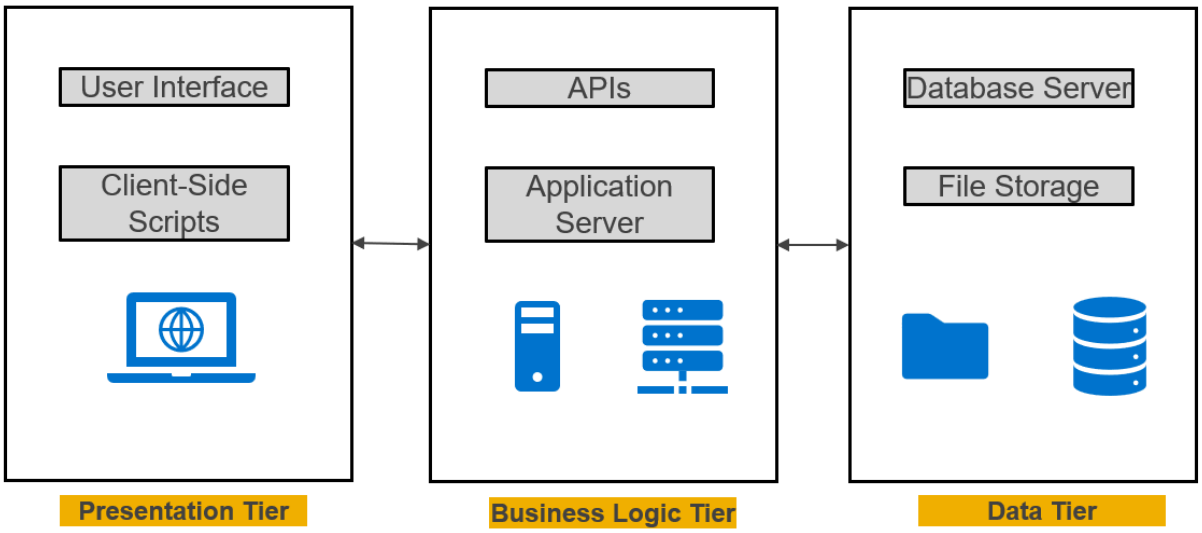


Fig 6: Multi-tier architecture design

Figure 6 illustrates the functions of each layer in a multi-tier architecture. In a typical multi-tier architecture design, there are the following tiers:

- **Presentation Tier** (Client Tier): This layer handles user interaction and interface rendering. It encompasses the User Interface (UI) and client-side scripts.
- **Business Logic Tier** (Application Tier): The core business rules reside here. It processes data received from the presentation tier. An Application Server typically hosts this layer.

- **Data Tier** (Database Tier): This tier is responsible for storing and retrieving data and connecting it to database servers and file storage.

Remember that layers refer to logical code separations, while tiers define the physical structure where these layers are deployed. The distinction is essential for understanding software architecture.

Data Flow diagrams: Review data flow diagrams that contain transactional data flows, data transformations, data storage locations, and data exchanges between system components and users. It provides a structured view of data flows, highlighting data inputs, outputs, transformations, and data dependencies critical for transaction processing.

While analyzing the data flow, it's essential to understand the data access methods. This involves thoroughly analyzing the system architecture, code base, and data flow patterns. Below are some common data access methods:

1. Application Programming Interfaces (API)
2. Direct interaction with databases or unstructured data stores
3. Others, e.g., file-based interfaces, message queues, or event-streaming platforms

API documentation: This documentation created by the application team can be used to understand how external systems or users interact with the business information system, including data exchanges, service invocations, and integration points.

In addition to architecture diagrams, flowcharts, sequence diagrams, use case diagrams, and business process modeling diagrams can be consulted to gain a deeper understanding of transaction flows. DNS flow logs may also be relevant depending on the specific context.

Constructing the Interaction Summary:

1. Determine transaction entry and exit points within and outside the business information system. These could be services, user interfaces, APIs, or other interaction points.
2. Use the flow diagram to trace the flow through the system. Identify the services and processes for handling transactions and their interactions with other entities and data.
3. Identify the DAAS elements accessed and their roles in the transaction using the architecture diagram.
4. Use the network diagram to identify how the transaction's access to DAAS elements is designed (Identify protocols, channels, and payloads).

The approach enables the documentation of the transactions. Documentation may have visual representations to improve readability, communication, and understanding. Review these flows with relevant stakeholders for clarity, completeness, and pertinence.

Leverage Scanning and Monitoring Tools

Organizations use one or more tools to discover and map their devices, services, and information flows. They identify all active connected devices, ports, active hosts, operating

systems, and traffic for a business information system, which may not directly map transactions; however, you can use a combination of information collected from these tools to gain insights into transaction flows. Ensuring proper placement and coverage of your scanning and monitoring tools ensures a comprehensive insight into your environment. Here, we have compiled a list of a few tools organizations use for various purposes and discussed how they can be used for transaction mapping.

Tool types	Purpose	Data	Usage
Application Observability and Analysis Tools (Example: Jaeger, Fluentd, Prometheus, OpenTelemetry)	These tools can understand the system from its external outputs.	1. Tracing data 2. Performance Metrics	Observability tools collect telemetry data that is consumed by analysis tools, which aid in querying or visualizing the data
Log Analysis Tools (Example: Splunk, ELK Stack, Sumo Logic)	These tools collect, centralize, and analyze log data from applications, servers, and network devices.	1. User activities (Authentication, access requests, data modifications) 2. Log events (App logs, server logs, security logs)	Help identify application errors, exceptions, and issues.
API Monitoring Tools (Example: Postman, Apigee, Kong, Layer7 API gateway)	These tools track and monitor interactions between applications, APIs, microservices, and external systems.	1. API calls (request, response, headers, parameters) 2. API status 3. Usage analytics (API usage patterns, API traffic)	It helps monitor API availability, track API usage, and identify API usage trends.
Network Monitoring Tools (Example: Wireshark, Nagios, SolarWinds Network, Service Mesh tools, Service Proxy tools, Identity Aware Proxies, Performance Monitoring tools)	They capture and analyze network traffic, including application-level protocols, to monitor system interactions and data exchanges.	1. Network traffic data (packet headers, payloads, protocols) 2. Network device status 3. Service-level communication (HTTP, FTP, SMTP, etc.)	It helps visualize network traffic flows, identify system communication patterns, and monitor service availability.
Database Monitoring Tools (Example: SolarWinds Database Performance Analyzer, Oracle Enterprise Manager, SQL Sentry)	These tools monitor database performance, query execution, and data access patterns to capture interactions with database systems.	1. Database server logs and events 2. Performance metrics 3. Session information	It helps analyze database transactions, data access patterns, and resource utilization.
Application/Transaction Monitoring Tools (Example: Dynatrace Transaction Monitoring, AppDynamics Business iQ, Service Mesh)	They track end-to-end transactional workflows, identify transaction paths, and monitor transaction performance.	1. Transaction traces 2. Business transaction data (user actions, system interactions) 3. Transaction performance metrics	It helps to analyze user interactions and the business impact of transactional activities.
Container Monitoring (Calico/Cillium, Network Policies, Runtime Monitoring, proxy mirrors)	These tools track container traffic, transactions, and interactions within the container ecosystem.	1. Transaction Traces 2. Network traffic data type 3. Service definitions and labels	It helps with mapping DAAS transactions and dependencies

Handle Encrypted Traffic

Encryption can make mapping transaction flows difficult because it can hide transaction content. While some tools can handle encrypted traffic, not all can, so workarounds are required. Some of these are as follows:

1. **Metadata Analysis**: Even with encrypted traffic, you can still analyze metadata such as packet sizes, timing, and flow directions. This can reveal patterns and anomalies without decrypting the content.

2. **TLS Inspection**: In controlled environments, organizations might implement TLS inspection. This allows for decryption and traffic analysis but requires careful management of security and privacy concerns.
3. **Encrypted Traffic Classification**: Machine learning techniques can classify encrypted traffic without decryption. For example, you could train a model to distinguish between different types of encrypted traffic based on flow characteristics.
4. **Application-Layer Visibility**: Analyzing application logs and telemetry can provide insights into transactions without decrypting network traffic. This approach focuses on the application layer rather than the network layer.

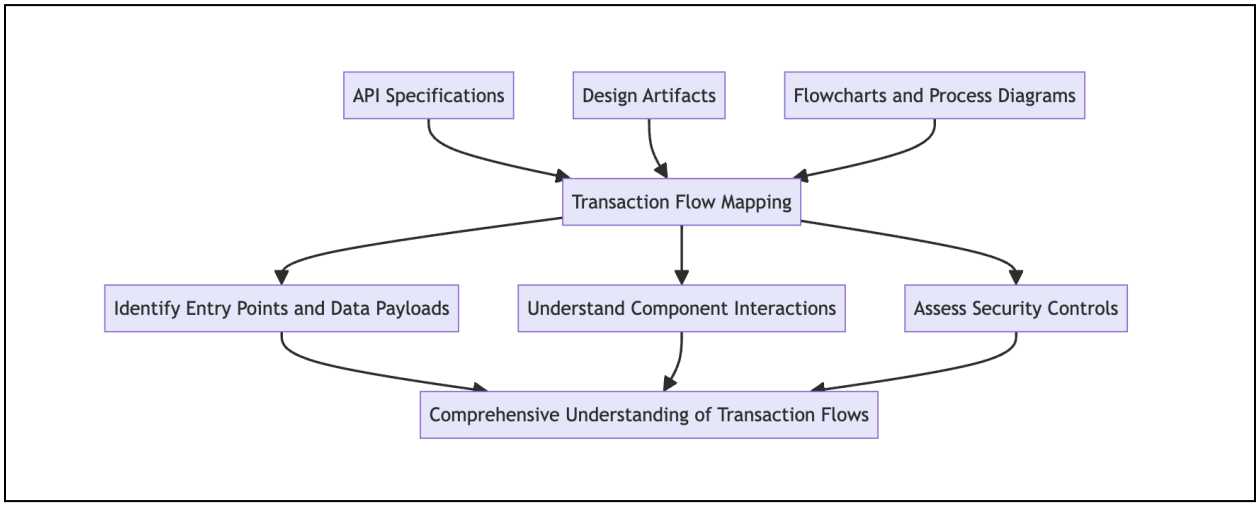


Fig 7: Transaction flow mapping process

Figure 7 shows an iterative process for using different documents, such as flow charts, API specifications, and so on, to create translation flow mapping. This aids in understanding business transactions and further refines the understanding to create comprehensive transaction flows.

Credit Card Business Information System Example

➢ 2.3 Identify Transactions.
Identify different types of transactions within and outside the system, such as data, financial, user, API, and system-to-system transactions.

Interaction entities	From: User/App	Transaction details	Communication Parameters (Protocols, Method, Format, Encryption, etc.)	To: App / Protect Surface / External service/User	Authentication	Service
User interaction	Customer	New Application - Customer card application details include PII	HTTPS, Asynchronous, JSON, TLS	Credit card web app	User credentials	Card application
	Sales Rep	Bill pay - Card details and billing information	HTTPS, Asynchronous, JSON	Credit card web app	User credentials	E-Payment transaction

	Customer	Cash withdrawal - Card details	XML/JSON, EMV Chip Authentication	ATM	ATM Pin	Payment transaction
Within Protect Surface	Credit card web app	New app decision - Customer card application details include PII	HTTPS, RESTful API, JSON, OAuth2.0, TLS	Underwriter app	API keys	Card decision
External interaction	Underwriter app	Request Credit check - Customer details, including PII	HTTPS, RESTful API, JSON, OAuth2.0, TLS	Credit agency	OAuth, JWT	Credit check API
	Underwriter app	Card print - Customer details including PII, Card details	HTTPS, RESTful API, JSON, TLS	Card Network agency	SAML	Card print service
	Underwriter app	Send credit report - Customer details including PII, Delinquent details	HTTPS, RESTful API, XML, TLS	Federal	SAML	Regulatory report
Other Protect Surfaces	Mortgage	Verify Loan details - Customer PII, Mortgage loan account details	SOAP, JSON, JSON Web Tokens	Underwriter app	API keys	Loans service
	Deposit	Verify savings - Customer PII, Savings account details	REST, XML	Underwriter app	API keys	Savings service
	Credit card web app	User authentication - Customer login credentials	HTTPS, OpenID Connect, RESTful APIs	Identity app	API keys	Authentication service
	Underwriter app	Send Email and Message - Card details, Transaction summary, alerts	HTTPS, RESTful API, JSON, TLS	Users	API keys	Email & message service

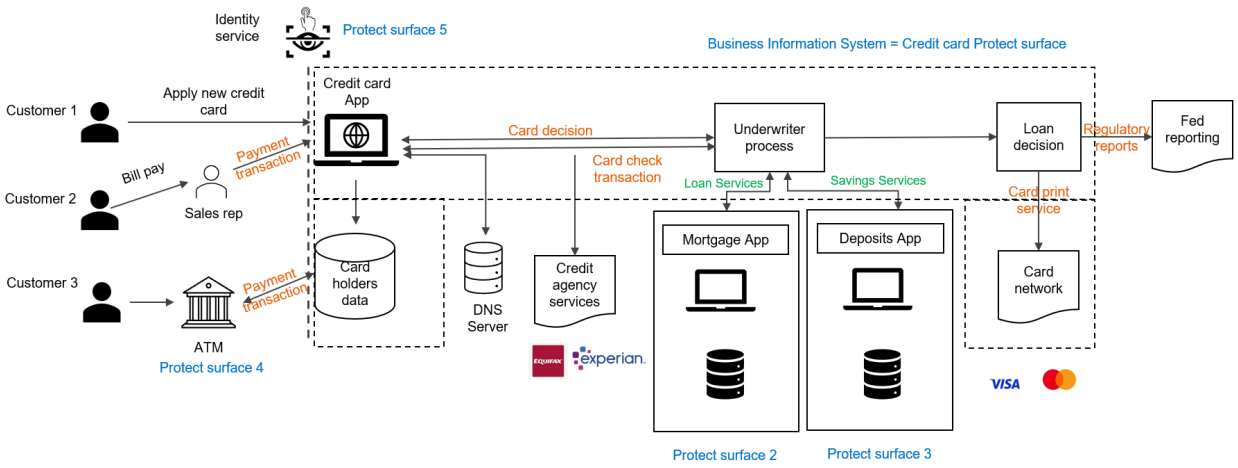
Also, identify the transactional touchpoints within each enterprise interaction, such as Active Directory service, logging, forensic service, IPS and IDS, and so on, where data is exchanged, processes are triggered, or services are invoked. Effective network scanning requires a thorough understanding of the scanning tool, the desired outcome of the scan, and the data or information that needs to be captured. This data is then analyzed to create a current network transaction flow map. A peer review process involving relevant stakeholders ensures the accuracy and approval of this map.

Detailed Architecture & Data Flow Documentation

By correlating the information derived from current and accurate architecture diagrams, network diagrams, data flow diagrams, network monitoring tools, log analysis tools, and their asset & configuration management systems organizations can create comprehensive data flow and architectural documentation for their business systems to facilitate enhancing system security. See the following illustration for an example.

Credit Card Business System Example

- Document system architecture and map data flows showing system interactions and process flows.
- Leverage the information gathered from internal and external interfaces and data flows to document system operations and architecture.



Validate and Refine Risk Classification & Security Maturity

As we learn about the nature of transactions, transaction types, and transactional workflows, we can leverage the information to perform security assessments of Protect Surfaces to ensure that when we deploy our ZT policies, the architecture has appropriate controls (policy enforcement points) in place for each interaction of DAAS elements based on the transaction flow mapping. This approach helps identify potential gaps in the security implementation and will act as a guide to identify weaknesses and develop a comprehensive Zero Trust architecture.

Organizations can use CISA's Zero Trust Maturity Model (ZTMM) to help assess the security maturity of their technology environment and business systems. The ZTMM helps assess current capabilities and practices across the ZT pillars and cross-cutting capabilities that provide integration, visibility, and interoperability. Each area is evaluated across four maturity stages: Traditional, Initial, Advanced, and Optimal. By assessing across the pillars, organizations can gauge their current maturity levels and identify areas for improvement.

- Map the DAAS elements and their interactions to the five pillars of CISA's ZTMM: Identity, Devices, Networks, Applications and Workloads, and Data.
- For each pillar, evaluate the current security policies and practices against the ZTMM's four maturity levels to identify strengths and weaknesses within the Protect Surface.
- Analyze the transaction flows to spot potential security gaps or risks, especially at interaction points between DAAS elements and external services. Compare these against the ZTMM recommendations for each maturity level.

By combining detailed interaction information from architecture and transaction flow documentation with the structured assessment framework provided by CISA's Zero Trust Maturity Model, organizations can gain a comprehensive understanding of their Protect Surface's security posture, identify specific areas for improvement, and develop a roadmap for enhancing their ZT implementation. Numberline Zero Trust security assessment is a sample tool that helps organizations assess their ZT maturity against CISA's maturity model.

The transaction flow mapping process can vary significantly depending on the environment, such as cloud, OT/IoT, or traditional IT. By tailoring the mapping process to each environment's specific characteristics and requirements, you can better understand how data flows and ensure robust security and operational integrity.

Mapping Transaction Flows for Cloud Applications

Choosing the Protect Surface, identifying its users and dependencies, and mapping transaction flows for a business information system in the cloud involve the same structured approach discussed above.

Protect Surface:
The cloud Protect Surface specifics depend on the organization's cloud infrastructure and the type used for each component. For example, an IaaS virtual machine is different from a PaaS function (e.g., lambda function) vs. hosted K8s workload vs. SaaS API, based on which the criticality of the assets and the desired level of control can be placed. Here are some examples.

Protect Surface	Data	Application	Assets	Services
Cloud Database Account	Customer records, financial transactions, analytics data	Database management systems, data encryption applications	Databases, tables, indexes	Database service (e.g., Amazon RDS, Azure SQL Database, Google Cloud SQL), backup and restore services, data replication services
Cloud Application Deployment Platform	Deployment scripts, configuration files, application source code	Continuous Integration/Continuous Deployment (CI/CD) tools, container orchestration platforms	Deployment pipelines, container images, build artifacts	Deployment services (e.g., AWS CodeDeploy, Azure DevOps), container services (e.g., AWS EKS, Google Kubernetes Engine)
Cloud Networking Service	Network traffic data, routing tables, firewall rules	Virtual Private Cloud (VPC) configurations, VPN gateways	Subnets, network interfaces, load balancers	Networking services (e.g., AWS VPC, Azure Virtual Network, Google Cloud VPC), DNS services, firewall services

Identify Dependencies and Map Transactions:
For the Protect Surface in the cloud, mapping transactions needs understanding internal and external dependencies, which includes identifying:

- The business information system utilizes all cloud services and resources, such as cloud storage, databases, compute instances, networking services, and APIs
- Third-party applications, services, and APIs integrated with the system, including SaaS applications, external databases, payment gateways, and identity providers
- Dependencies on underlying infrastructure components, cloud regions, availability zones, virtual networks, load balancers, and security services

Along with leveraging existing documents such as system architecture diagrams, data flow diagrams, transaction flow diagrams, and dependency matrices of the cloud, organizations can take advantage of cloud monitoring and analytics tools provided by cloud service providers (e.g., AWS CloudWatch, Azure Monitor, Google Cloud Monitoring) to monitor system interactions, resource utilization, network traffic to map transaction flows. One can also enable logging and monitoring features on their cloud service provider environment to gain visibility into cloud-based application traffic. These features may include cloud-specific logging services that capture information about network traffic, access, and other relevant activities. Cloud service providers may offer network flow logs that provide detailed information about network transactions, source and destination IP addresses, ports, and network protocols carrying the payload. Third-party solutions operating on the cloud environment may offer similar or advanced mapping and monitoring capabilities running on the cloud environment. By enabling and analyzing these logs, organizations can better understand transaction flows.

Mapping Transaction Flows for Operational Technology (OT) and Internet of Things (IoT)

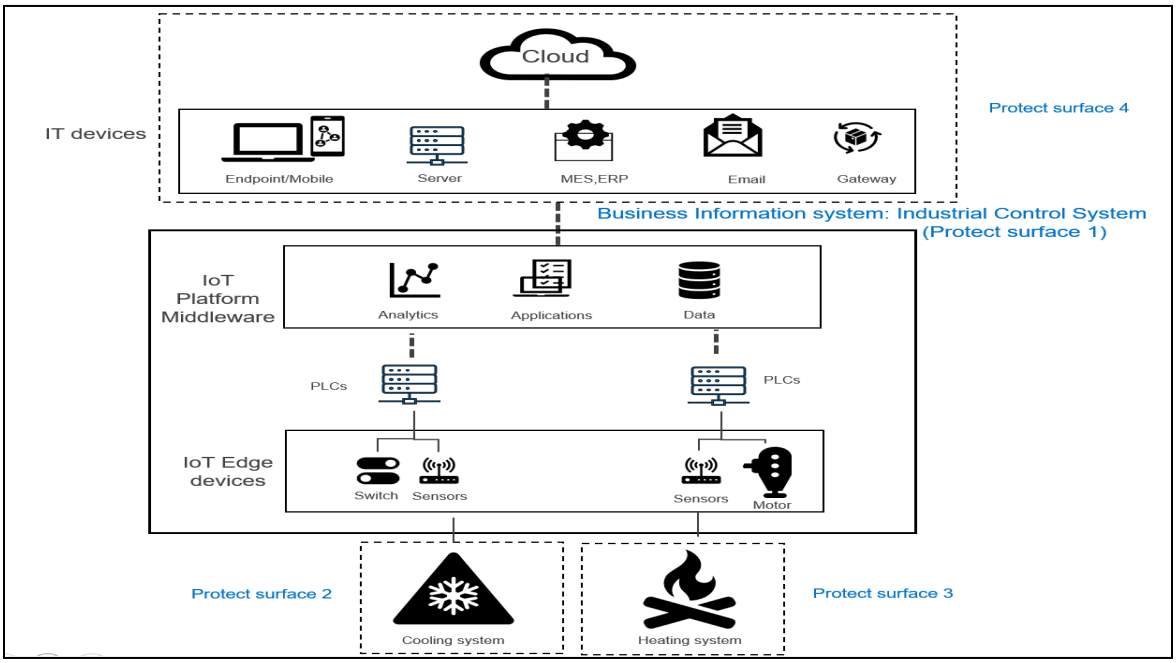
Mapping transaction flows within an OT/IoT environment requires a specialized approach due to its unique challenges and considerations. A thorough understanding of the environment is needed, including the types of devices, sensors, actuators, control systems, protocols, and data flows involved.

The Protect Surface could be an industrial control system (ICS), supervisory control and data acquisition (SCADA) system, Building Automation System like HVAC, programmable logic controller (PLC), or any other critical business information system in the OT/IoT environment. While identifying DAAS elements and their dependencies for the OT/IoT system, ensure that interactions of Edge devices, the Middleware platform, and IT systems/processes are included:

- **Sensors/Actuators** (Edge Devices): Physical sensors and actuators in the operational environment.
- **IoT Gateways** (Edge Node): Edge computing nodes for protocol translation, data filtering, and local processing.
- **IoT Platform** (Control Plane): Central IoT platform handling device management, data ingestion, processing, and control.

- **Cloud/Remote Services**: Cloud platforms or remote servers provide monitoring, analytics, and management services.

Let us consider another fictitious example of an Industrial Control System in a chemical plant, which uses sensors and PLCs to manage chemical processes.



Business Information system		**Industrial Control System**
Step 1: Identify the Protect Surface & DAAS	Data	Control, sensor, and process data used to manage chemical processes in a chemical plant
	Applications	Production chemical process control
	Assets	Chemical plant sensors and PLCs
	Services	Heating, Ventilation, and Air Conditioning (HVAC)
Step 2.1: Identify Users		System Administrators, Control Engineers, Maintenance Technicians, Quality Control (QA/QC) Personnel, External Service Providers
Step 2.2: Identify dependency		Integration with IoT platforms, Enterprise Systems, cloud services, HVAC systems
Step 2.3: Identify Transactions		1. Transmission of sensor data to PLCs 2. PLC's executes control commands to Motors, switches 3. Services for alarm management 4. Services to trigger HVAC systems 5. Services to Integrate with enterprise systems such as ERP (Enterprise Resource Planning), MES (Manufacturing Execution Systems)

Refer to *Zero Trust Guidance for Critical Infrastructure* for detailed guidance for implementing a ZT strategy for OT/ICS environments.

Challenges in Aggregation of Transaction Flows

In the digital landscape, data traverses multiple regions and platforms, making it imperative for organizations to grasp transaction flow across these diverse environments. This flow requires integrating transaction data from on-premise, cloud-based systems, and OT/IoT devices into a unified view of the transaction flow. A unified view provides comprehensive visibility into the enterprise network.

The key to seamless integration lies in employing common data standards and conventions across these systems to enhance interoperability. Middleware solutions can effectively translate data formats, bridging the gap between disparate systems and enabling seamless data exchange. Organizations can better understand their operational dynamics by mapping complete transactions, identifying potential security risks, and implementing targeted mitigation strategies.

Moreover, maintaining data integrity and consistency during aggregation is critical, as discrepancies can lead to incorrect access decisions or security vulnerabilities within the Zero Trust framework. To address these challenges, organizations must invest in advanced data management and orchestration tools that can seamlessly aggregate and harmonize transaction flows while upholding the principles of Zero Trust.

How Automation and AI/ML Can Assist Transaction Flow Mapping

One of the challenges in transaction mapping is modern operating environments' complexity and dynamic nature. Frequent system configurations, software changes, and new third-party integrations lead to outdated documentation and incomplete transaction mapping. To address this, automation and artificial intelligence/machine learning (AI/ML) tools can help keep documentation and transaction mappings current and accurate, ensuring that they reflect the evolving landscape of the environment.

Complex environments generate large volumes of data, making annual analysis costly, time-consuming, and error-prone. AI/ML technologies are adept at handling vast amounts of data and complex dependencies, efficiently analyzing and extracting relevant information from diverse logs, network traffic, and other sources. This capability provides a more accurate and comprehensive understanding of transaction flows.

Here are some ways AI and ML can be leveraged in transaction mapping:

- **Dynamic Analysis and Adaptation**: AI and ML algorithms can automatically map interactions between users, devices, applications, and systems by analyzing network traffic, log data, and application traces. They also continuously update transaction flows in real time as new devices, services, and users interact with the network, ensuring that changes in the environment are reflected in the transaction flow documentation.

- **Pattern Recognition**: Machine learning models can identify transaction flow patterns, helping categorize and group similar interactions. This can simplify the mapping process by highlighting common pathways and anomalies.
- **Automated Discovery**: AI-powered tools can automatically discover and map network resources, applications, and interactions, reducing the manual effort required in complex environments.
- **Predictive Analysis**: ML models can predict potential changes in transaction flows based on historical data, helping organizations proactively adjust their ZTA.

Below are some examples of tools enabled with AI/ML technologies that can be used:

- **Cisco Secure Workload** uses machine learning to assist in Automated application discovery and dependency mapping.
- **Illumio** uses AI to map and visualize transaction flows between workloads, applications, and users.
- **Zscaler Zero Trust Exchange** employs AI to monitor user-to-app and app-to-app transaction flows across cloud and on-premises environments.

Refine Protect Surface(s) based on Transaction Flow Mapping

Transaction flow mapping unveils a comprehensive view of data flows, enabling identifying all users and DAAS elements. This detailed insight empowers organizations to analyze the transaction flows of access to DAAS elements and pinpoint those that are not understood or misunderstood, redundant, deprecated, or no longer essential for business operations. Unnecessary or unwanted protocols and services within the network pose significant security risks that warrant immediate elimination.

Figure 5 illustrates how transaction flow mapping is used to understand and refine the initially defined Protect Surface:

Defined Protect Surfaces: These Protect Surfaces are defined in Step 1 of the ZT implementation process based on initial understanding. The diagram illustrated in Figure 5 shows web servers, web applications, cloud applications, email services, cloud services, and App interface services.

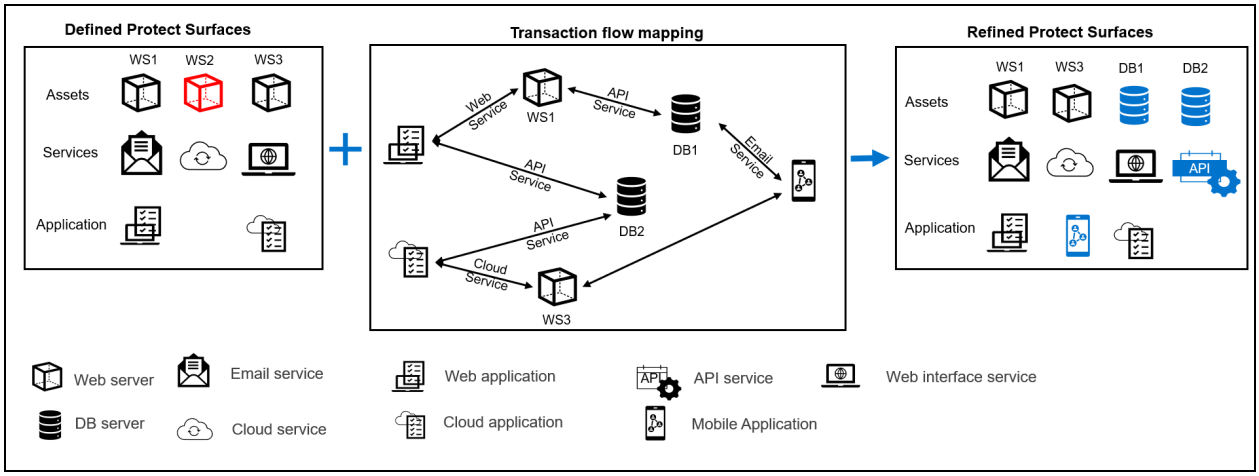


Figure 5: Refining the Protect Surface based on Transaction mapping

Transaction Flow Mapping: As mentioned in the above sections, the transaction flow mapping of Protect Surface is created based on various techniques such as comprehensive analysis of application architecture, data flow diagrams, user interaction diagrams, stakeholder interviews, etc. The diagram illustrates how applications interact via various services to the servers and databases. It is evident that only two web servers out of 3 are used, 2 DB servers are used, email services are used by mobile applications for interface, and API services are used for communications.

Refined Protect Surfaces: Based on the insights derived from transaction flow mapping, the Protect Surface is refined by removing unnecessary DAAS elements and adding missing ones. However, before disabling any critical service, organizations need to take a series of cautionary steps to avoid disruptions, security issues, and data loss. Below are some best practices:

- ➔ Identify and engage relevant stakeholders (e.g., IT teams, security, compliance, business owners, and application owners) to understand the impact of the service on their operations.
- ➔ Perform an impact assessment and dependency mapping to understand which systems, applications, and processes rely on the service being disabled.
- ➔ Review at least one year of transaction logs, usage patterns, and monitoring data to determine how frequently the service is used and whether it plays a critical role during peak periods or in specific workflows.
- ➔ Assess the security impact of disabling the service, such as exposing systems to vulnerabilities, compromising access control, or breaking critical security mechanisms.
- ➔ Develop a rollback plan to revert changes quickly if the disabling of the service leads to unexpected issues.

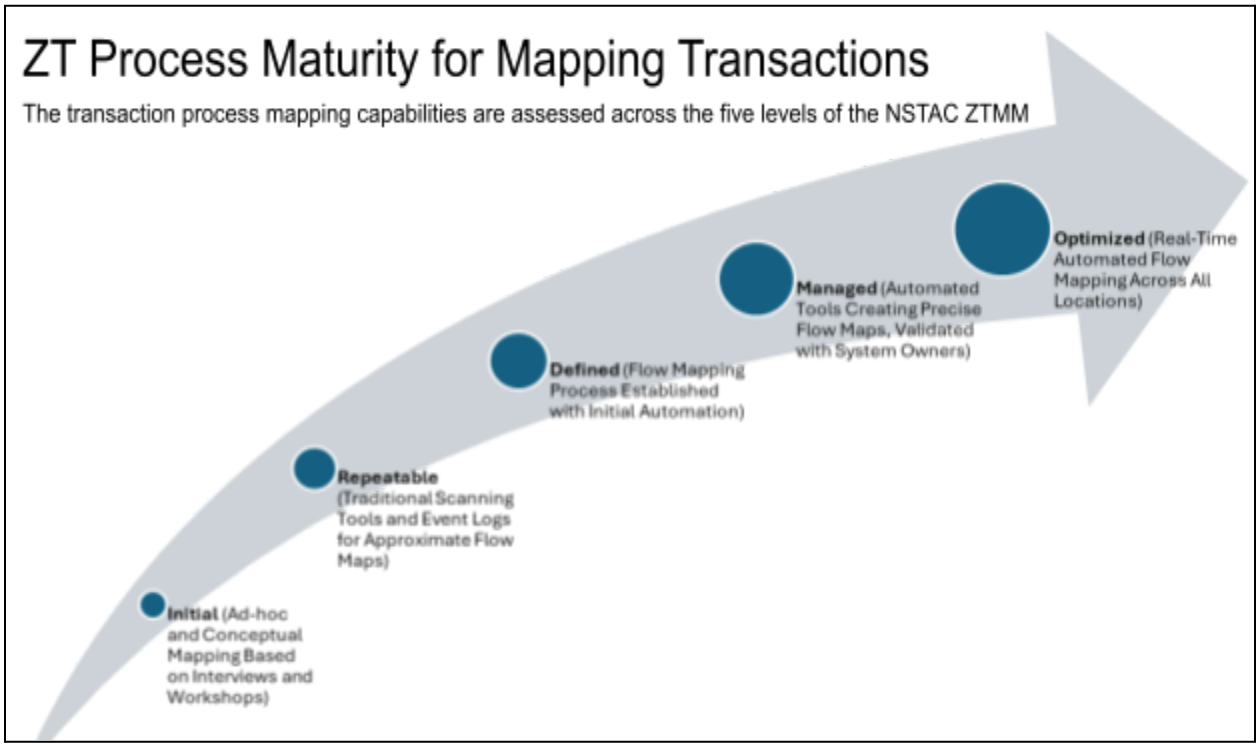
In Figure 5, based on transaction mapping flow, web server 2 was eliminated (highlighted in red), DB servers were added, API service was included, and a missing Mobile application (highlighted in blue) was also included.

Organizations effectively reduce the attack surface by minimizing the number of exposed services while establishing a well-defined and rigorously protected perimeter. Automating the

refinement process ensures continuous network monitoring, enabling proactive detection and mitigation of instances where removed or restricted services or protocols resurface. This ongoing vigilance safeguards the organization from potential security breaches and ensures that the attack surface remains minimized over time.

Transaction Flow Mapping Process Maturity Model

The Transaction Flow Mapping Process Maturity Model (TFMPMM) presented is a framework for assessing and improving the maturity of an organization's transaction flow mapping process. The model uses five maturity levels: Initial, Repeatable, Defined, Managed, and Optimized. Each level describes the key characteristics of an organization's transaction flow mapping processes, tools, and outcomes.



1. Initial (Ad-hoc and Conceptual Mapping Based on Interviews and Workshops)
At this stage, organizations rely on informal processes to conceptualize transaction flows. Flow maps are created based on interviews, workshops, or brainstorming sessions. No formalized or repeatable process is in place, and the maps are often incomplete and lack detail. The expected outcomes at this stage are a basic understanding of transaction flows, but the mapping process is fragmented and reactive. Maps are not regularly updated, and there is little integration with security or operational frameworks.
2. Repeatable (Traditional Scanning Tools and Event Logs for Approximate Flow Maps)
Organizations have started using traditional scanning tools like Nmap or Wireshark and event log analysis tools like Splunk to construct flow maps. The process becomes more repeatable and somewhat structured, though the accuracy and coverage of the maps are still limited. The

flow mapping process can be repeated for different parts of the system, but it still relies on manual effort for interpretation and analysis. The flow maps at this level are more repeatable and standardized, though still approximate. The process is more consistent but lacks precision.
3. Defined (Flow Mapping Process Established with Initial Automation)
A defined process for transaction flow mapping is now established. Automated tools, such as Nagios, Wireshark, and so on, begin to be deployed, allowing for more structured data collection and analysis. The process is now formalized and integrated into security and operational practices, though manual refinement is still needed. The expected outcomes at these levels are more accurate flow maps that cover a wider range of transactions. The mapping process is formalized and integrated into broader security frameworks, but validation and some gaps may still exist.
4. Managed (Automated Tools Creating Precise Flow Maps, Validated with System Owners)
At this stage, the organization uses advanced automated tools such as AppDynamics, Dynatrace, and so on, to create precise flow maps. These maps are validated with system owners to ensure their accuracy. The process is proactive and managed, meaning that the flow maps are comprehensive, up-to-date, and fully integrated with risk management and security processes.
5. Optimized (Real-Time Automated Flow Mapping Across All Locations)
The organization has reached the highest level of maturity, where transaction flows are automatically mapped across all locations and updated in real time. Real-time analysis tools powered by AI/ML, such as Illumino, Ciso Secure Workload, and so on, allow for immediate identification of abnormal or unauthorized activities, and security measures are automated based on flow anomalies. Flow maps are always current, adaptive, and used to dynamically enhance the organization's security posture.

Benefits of Mapping Transaction Flows

By following this document's guidance and best practices, organizations can effectively map their transaction flows, gain visibility and insights, and design a ZTA that aligns with their specific security requirements and business objectives.

The insights gained from mapping transaction flows form the foundation for implementing effective security controls, such as granular access policies, network segmentation, and strategically placed security solutions. By understanding how data flows within and across applications and workloads, organizations can identify potential risks, detect anomalies and threats in real time, and respond quickly to security incidents.

The benefits of mapping transaction flows extend beyond security. By understanding the interactions and dependencies within their application and workload environment, organizations can optimize performance, identify bottlenecks, and make informed decisions about capacity planning and resource allocation. This complete view of the system enables organizations to align their IT operations with business objectives, enhance user experiences, and drive innovation.

Here are some of the benefits that can be gained:

Visibility and Understanding: It provides a comprehensive view of the organization's digital interactions, helping security teams understand how data moves within the environment.

Policy Enforcement: With a detailed map of transactions, organizations can enforce security policies more effectively, ensuring only authorized transactions occur.

Incident Response: In the event of a security incident, having a mapped understanding of transactions helps in tracing the source, understanding the extent of the breach, and responding more effectively.

Enhanced Security Posture: Improved visibility into transaction flows allows organizations to better understand the dynamics of network communication. This visibility enables more effective security measures, reducing the risk of unauthorized access and security breaches.

Risk Reduction: Mapping transaction flows facilitates identifying and eliminating unnecessary services, protocols, and paths. This reduction in the attack surface minimizes vulnerabilities and lowers the risk of cyber threats.

Compliance Adherence: Understanding transaction flows supports compliance efforts by ensuring access controls align with regulatory requirements.

Operational Efficiency: Transaction flow mapping helps organizations identify business transaction paths. This allows more efficient network segmentation and access control policies.

Data Protection: Understanding transaction flows aids in the identification and protection of sensitive data. Organizations can implement data-centric security measures based on the knowledge of how data moves within the network, reducing the risk of data breaches.

Business Continuity: A well-mapped and segmented network based on transaction flows improves business continuity. In the face of cyber threats or incidents, critical business operations can continue while minimizing the impact on the organization.

Customer and Partner Trust: Demonstrating commitment to cybersecurity through effective transaction flow mapping builds trust with customers and partners and can be a competitive advantage.

The Next Step: Building the Zero Trust Architecture

Building a Zero Trust Architecture (ZTA) is the third step in the Zero Trust Implementation method. It involves architecting a ZT environment with policy enforcement points (PEPs) tailored to the Protect Surface(s) defined in Step 1 and transaction flows mapped in Step 2. The design will be based on current (observed) and desired (future state) transaction flows from identities, using devices, with traffic moving across networks to, from, and within each Protect Surface component—a transaction flow mapping document aids in providing this information.

Implement Micro-Segmentation
Transaction flow mapping helps organizations plan and implement an appropriate level of micro-segmentation by dividing the network into smaller, isolated segments, often at the individual workload, application, or device level. By understanding transaction flows, you can segment the network based on logical groupings of resources and applications, reducing the attack surface. Based on transaction flows, the organization can define appropriate access policies for each Protect Surface component, enforced by the PEPs. Note that it may be sensible for some organizations to Protect Surfaces by starting with coarser-grained macro segmentation and then improving and evolving to microsegmentation over time.

Context-based Access Control
In a ZT model, access control is dynamic and context-aware. Transaction flow mapping provides an essential element of context that will be used within access policies to make access decisions based on user identity, device posture, location, and the specific transaction being performed.

Least Privilege Access
Transaction flow mapping supports the principle of least privilege access by identifying which transactions are essential for specific roles or applications. This information is crucial for restricting access to the minimum necessary to perform tasks. Transaction flow mapping can establish a baseline for normal network behavior. Any deviations or anomalies in transaction flows can trigger alerts and security responses, enhancing security incident detection and response capabilities.

Encryption and Data Protection
Ensure the adoption of strong encryption protocols to protect the transaction flows and implement data loss prevention mechanisms. Transaction flow documents comprehensively outline the devices involved in a transaction, the communication protocols used by each device, and the sequence of data exchanges between them.

Security Monitoring and Logging

- Implement centralized logging to capture and analyze system logs for security events.
- Deploy security information and event management (SIEM) solutions for real-time monitoring and correlation.

While these examples provide a glimpse into the security controls developers can incorporate into their designs, a more comprehensive exploration of design considerations awaits readers in 'Building a Zero Trust Architecture,' the third installment in the Zero Trust Implementation series.

Conclusion

This in-depth guidance is designed for ZT implementers and explores the complexities of interactions between users and DAAS elements in the business system/Protect Surface and related network resources. It extends beyond conventional cybersecurity practices to provide insights into creating comprehensive system architecture and data flow mapping documentation, arming Zero Trust architects and implementers with the information to cost-effectively fortify their organization's cybersecurity posture, one Protect Surface at a time.

This document also offers guidance for navigating the complex terrain of understanding system architecture, transaction flows and security maturity within the Zero Trust paradigm, facilitating enhancing cybersecurity resilience and establishing a foundation for establishing and enforcing adaptive and robust security policies tailored to the unique characteristics of the organization's business system operations and imperatives.

Useful References

NSTAC

- NSTAC Report to the President on Zero Trust and Trusted Identity Management

John Kindervag Circle recordings and documentation

- Zero Trust Implementation and Guiding Principles Briefing by John Kindervag (Passcode: ZTimplement101!)
- ZT Data Protection and Privacy Briefing by John Kindervag (Passcode: DataPillar7!)
- Financial sector recording
- ZT Summit 2023 recording
- Various slide decks: FDCI, DoD, ATARC

Palo Alto Networks

- The Zero Trust Learning Curve: Deploying Zero Trust One Step at a Time

CSA Zero Trust Advancement Center

- Zero Trust Advancement Center

NIST

- NIST SP 800-207 Zero Trust Architecture
- NIST SP 800-207A A Zero Trust Architecture Model for Access Control in Cloud-Native Applications in Multi-Cloud Environments
- Discussion Draft of the NIST Cybersecurity Framework 2.0 Core
- NIST SP 1800-35 Implementing a Zero Trust Architecture, figure 1 page 54, 2nd preliminary draft

DoD Reference Architecture & Strategy

- Department of Defense (DoD) Zero Trust Reference Architecture Version 2.0
- DoD Zero Trust Strategy

CISA Maturity Model V2

- CISA Zero Trust Maturity Model Version 2.0

NSA

- CIS: Advancing Zero Trust Maturity Throughout the Network and Environment Pillar

Books

- Project Zero Trust: A Story About a Strategy for Aligning Security and the Business by George Finney

Scanning tools:

Service Mesh - https://landscape.cncf.io/guide#orchestration-management--service-mesh

Service Proxy - https://landscape.cncf.io/guide#orchestration-management--service-proxy
Observability & Analysis Tools - https://landscape.cncf.io/guide#observability-and-analysis--observability

Container Network Policies:
Cillium: https://docs.cilium.io/en/latest/security/policy/index.html
Calico: https://docs.tigera.io/calico/latest/network-policy/adopt-zero-trust

Understanding ZT Maturity Model:
https://cloudsecurityalliance.org/blog/2023/05/17/understanding-the-two-maturity-models-of-zero-trust

Glossary

- CSA Glossary (main/primary)
- On2IT ZT Glossary - Zero Trust Dictionary (John Kindervag)
- CSA SDP Glossary (Software Defined Perimeter)