Summer 2022

Zero Trust Architecture: Framework and Case Study

Cody Shepherd
Boise State University

2022

Zero Trust Architecture

FRAMEWORK AND CASE STUDY

CODY SHEPHERD

Completed as part of the CORe 596 Independent Study, Summer 2022
Cyber Operations and Resilience Program
Boise State University

Introduction

The world and business are connected and a business does not exist today that does not have potentially thousands of connections to the Internet in addition to the thousands of connections to other various parts of its own infrastructure. That is the nature of the digital world we live in and there is no chance the number of those interconnections will reduce in the future. Protecting from the "outside" world with a perimeter solution might have been enough to reduce risk to an acceptable level in an organization 20 years ago, but today's threats are sophisticated, persistent, abundant, and can come from pretty much anywhere: a hostile nation-state, a hacker in their parents' basement, a disgruntled system admin, an unsuspecting accountant victim, and countless others. The methodologies and principles of perimeter-based protections of yesteryear must be constricted down to the resource level and constantly evaluated for risk in order to be effective today and into the future…and that constriction comes in the form of zero trust. We must no longer solely focus on protecting the perimeter of the network but take that same mentality and make that perimeter protection applicable to every user, resource, service, and asset within operation in our enterprise and our cloud services.

Greenfield organizations are able to architect zero trust principles into their systems from origin, giving them a faster path to full adoption and incorporation into their business processes. Unfortunately, existing enterprises do not have this luxury but must begin the journey of transitioning to this new realm of cybersecurity practice. This research paper is intended to manage expectations of implementing zero trust and what benefits can be derived from it and what risks can and will exist during and after implementation. I will create a common understanding of what zero trust really is, and what it is not, and explore a proposed framework that can be used for an organization to plan and implement zero trust. The term "zero trust" is a misnomer. There cannot be systems and communication (required for a digital business to operate) if there is no trust between systems sharing information. What the term "zero trust" is really saying is that there is no *implied* trust in data access

and communications. Transactions and access are constantly verified that the requestor is genuinely who they indicate themselves to be, and they are authorized to do what they are intending to do. To clarify understanding of the past paradigm to what we are referencing in zero trust, we can associate and reference a family and their home. The aforementioned perimeter protection was how our homes looked in 1950's television shows where the most protection a home needed was a lock on the front door. Everyone in the home was trusted, Mom was always home keeping an eye on things, the kids were good and could run freely around the house with full access to everything and there was no threats or impacts to anything inside the house, so long as the front door was locked and only Dad had the keys to the lock in order to traverse that perimeter.

In today's world, there are threats of "bad guys" coming in the windows, hiding in the basement for months, spy satellites overwatching activity, locks being picked, the kids might want to "explore" Dad's liquor cabinet or access unauthorized applications on the TV or computer, and some kids might have gone awry and decide they are going to slowly steal from the family. Keeping the modern home (i.e., corporate network) safe from all of these possible threats is to remove the incentive and opportunity from the adversaries and make it so they cannot accomplish their malicious goals, or, if they do start to succeed, that unauthorized activity is confined to the specific incident and has no chance of spreading throughout the house.

What is "Zero Trust"

The term "zero trust" is powerful and catchy and makes for a strong marketing term. In the past year, to my ears and what I hear in the cybersecurity realm, is that zero trust is the saving grace of all things cybersecurity. That is where all organizations need to get to as fast as possible if they want to be more secure and protected against threats than they have ever been and President Joe Biden even issued Executive Order 14028 requiring federal civilian agencies to establish plans to drive adoption of Zero Trust Architecture by 2024. To an outsider of cybersecurity but someone who happens to work in

the Information Technology field, zero trust is the holy grail of Security Architecture. But the term "zero trust" is much more than a marketing term and has a lot more substance behind the definition. NIST (National Institute of Standards and Technology) defines Zero Trust (ZA) and Zero Trust Architecture (ZTA) as follows [1]:

> *Zero trust* (ZT) provides a collection of concepts and ideas designed to minimize uncertainty in enforcing accurate, least privilege per-request access decisions in information systems and services in the face of a network viewed as compromised. *Zero trust architecture* (ZTA) is an enterprise's cybersecurity plan that utilizes zero trust concepts and encompasses component relationships, workflow planning, and access policies. Therefore, a zero trust enterprise is the network infrastructure (physical and virtual) and operational policies that are in place for an enterprise as a product of a zero trust architecture plan.

NIST Special Publication 800-207 on zero trust architecture is referenced in many of the research documents which contribute to my understanding of zero trust architecture I discuss in this paper and many of the principles and recommendations included within. I respect and hold in the highest regard as I reference them throughout my research, but I think their definition is a bit complex and would not create the right visual representation of zero trust and zero trust architecture to, for example, a company executive. The key words missing from these definitions are *implicit* and *explicit*. Zero trust "concepts and ideas" that NIST is implying in their definition mean zero *implicit* trust and using constant authentication and authorization decisions at every interaction point in the network to grant risk-appropriate, *explicit* trust. To me, that is a more streamlined, accurate, and impactful definition.

Implementing zero trust architecture is founded on the idea that every "other" asset, resource, or user is hostile and your system is already compromised. Every system and access to every bit of data must be protected as if it is under attack by the resource right next to it. In our home example, that liquor cabinet is not going to implicitly accept that the person opening the door is Dad just because Dad lives there and the house has locks on the front door. That liquor cabinet is going to make Dad verify he is "Dad" and provide multiple authentication factors (password, biometric, etc.) that will be verified by an independent third party (This will be the policy engine and policy administrator in the control plane

we will discuss later) before Dad gets access to the liquor cabinet. Drunk Uncle Earl and 9-year old Billy, even when they are within the perimeter of the house, do not meet the criteria necessary (the privilege) to be able to access the cabinet and their attempts will fail and an alert will be sent to Dad. In zero trust, access is never implied. It is only explicitly granted after a review of the authorization of the access being allowed and verification of the entity trying to access, and this review happens constantly to verify that the explicit access given at a certain timestamp is still prudent and relevant at time *n+timestamp*.

What is Not "Zero Trust"

In my organization I have personally heard use of the term "zero trust network" interchangeably with "network segmentation". Other cybersecurity analysts have described the principle of least privilege and called it zero trust and others have referenced Enterprise Security Risk Management synonymously as zero trust. In my research, all of these are components that can work together to achieve the goals of zero trust, but they are not conclusive and comprehensive to the degree of matching the definition I described above. Those items are singular components of our journey that will be discussed in this research and applied to the framework I am proposing for a company to build a program around adopting zero trust. Zero trust architecture is a huge set of design principles that are applied at multiple levels of the enterprises' systems: including aspects of identity management, encryption, access management, monitoring, data transport, sessions, and configuration management. For example, network segmentation is a form of intermediary filtering [2] where the traffic can flow between hosts within different vlan networks but must traverse a firewall for the East-West traffic and abide by the rules stipulated in those firewalls. This is a good contribution to zero trust architecture, but it still misses the host filtering component. So as long as the connection between the vlans meets the firewall parameters, the trust just became implied to the entire receiving network segment, and all of its hosts, and a key principle of zero trust was not achieved. Zero trust architecture is much more comprehensive and involved than looking at any single effort dealing with network equipment, server

hardware, applications, hypervisors, PKI (Public Key Infrastructure), IAM/IGM (Identity Access/Governance Management), Active Directory, or configuration management…it takes application of all of these working together.

There is no single tool or platform that can be installed by the enterprise to accomplish zero trust. For a technologist, this can be difficult to ingest. Surely someone has packaged up a suite of products they can market under one umbrella that can accomplish the goals of zero trust, right? No, that is not the case. It is important to remember that accomplishing zero trust is the application of design principles to accomplish a goal. The goal of any cybersecurity initiative should be to support and protect business operations. If it is not important to the business, then it should not be a cybersecurity investment. NIST SP 800-160 recommends using NIST SP 800-37 on Risk Management Framework as the steps to introduce zero trust architecture into an existing perimeter-based architected network [1], but I would take it one step further, as we will discuss in the next section as I develop my own zero trust architecture framework.

Getting Started

First, we begin with logistics and where we left off. Using the NIST Risk Management Framework (RMF) as a catalyst point to survey the assets, users, data flows, and business workflows of the organization. This will be the foundation of preparing the organization for zero trust implementation planning. The RMF steps of Prepare, Categorize, Select, Implement, Assess, Authorize, and Monitor may already be in effect in an organization independent of any zero trust initiative, but the foundations are very sound and extremely applicable to beginning the journey into zero trust. Gilman and Barth explicitly state that the first steps to implementing zero trust is to diagram all network flows [2], as they follow an RFC-style prioritization list for defining scope of zero trust networking which prioritizes that "All network flows MUST be authenticated before being processed" [2]. In my opinion, scoping the execution of the "what" (network flows) does not happen first. Zero trust is not a bolt-on solution to an

existing network, so planning an approach to systematically connect the design principles of zero trust to the goals of the organization is critical. I would recommend the NIST SP 800-160v2 [3] constructs approach to developing Cyber Resilient Systems. This approach is grounded in defining *Design Principles*, which is exactly what zero trust architecture is! By articulating the design principles of zero trust and then tailoring and mapping those to NIST-defined or self-defined later constructs, then a zero trust-specific framework can be constructed using this established model. The zero trust design principles influence *implementation approaches* (which is the "what" that Gilman and Barth were describing). Those approaches influence the *Techniques* which the organization plans to utilize. NIST has defined families for techniques in SP 800-160v2 but in developing our custom structure for zero trust architecture, the techniques would be the people, processes, and tools which will support the next layer up, the *Objectives*.

Zero trust is accomplished not through one suite of applications purchased through a vendor, it is the seamless integration and constant communication of a multitude of tools and systems all contributing their part to the architecture as a whole. The techniques section of this framework is where we ensure that we have the proper tools implemented (or planned), processes to support the implementation while keeping the business operational, and the people to make this all possible per the requirements of the design principles and to execute on the implementation approaches in support of accomplishing the objectives. Moving up the framework to the *Objectives* the techniques will support is where we will track our metrics and KPIs towards accomplishing our goals of zero trust. The goals of zero trust are directly influenced by the Business' Risk Management Strategy! We will have clear priority of the goals of the multi-year, incremental zero trust implementation effort because those exact goals are driven from the organizations' risk tolerance and strategy for managing risk.

How do we define the objectives to get from the execution of the techniques up to the goals? This answer lies in overlaying the CISA Zero Trust Maturity Model five pillars of implementation. The

pillars of Identity, Device, Network/Environment, Application Workload, and Data [4] all include various functions that can be rated for maturity as “Traditional”, “Advanced”, or “Optimal”. There can be diminishing returns for an organization to strive for “Optimal” of every function of every pillar (objective), but with clearly defined goals we will be able to determine the correct maturity level of each objective and measure our progress towards the company’s goals regarding zero trust. The marriage of these models to apply to initiating the implementation of zero trust architecture is displayed in Figure 1.

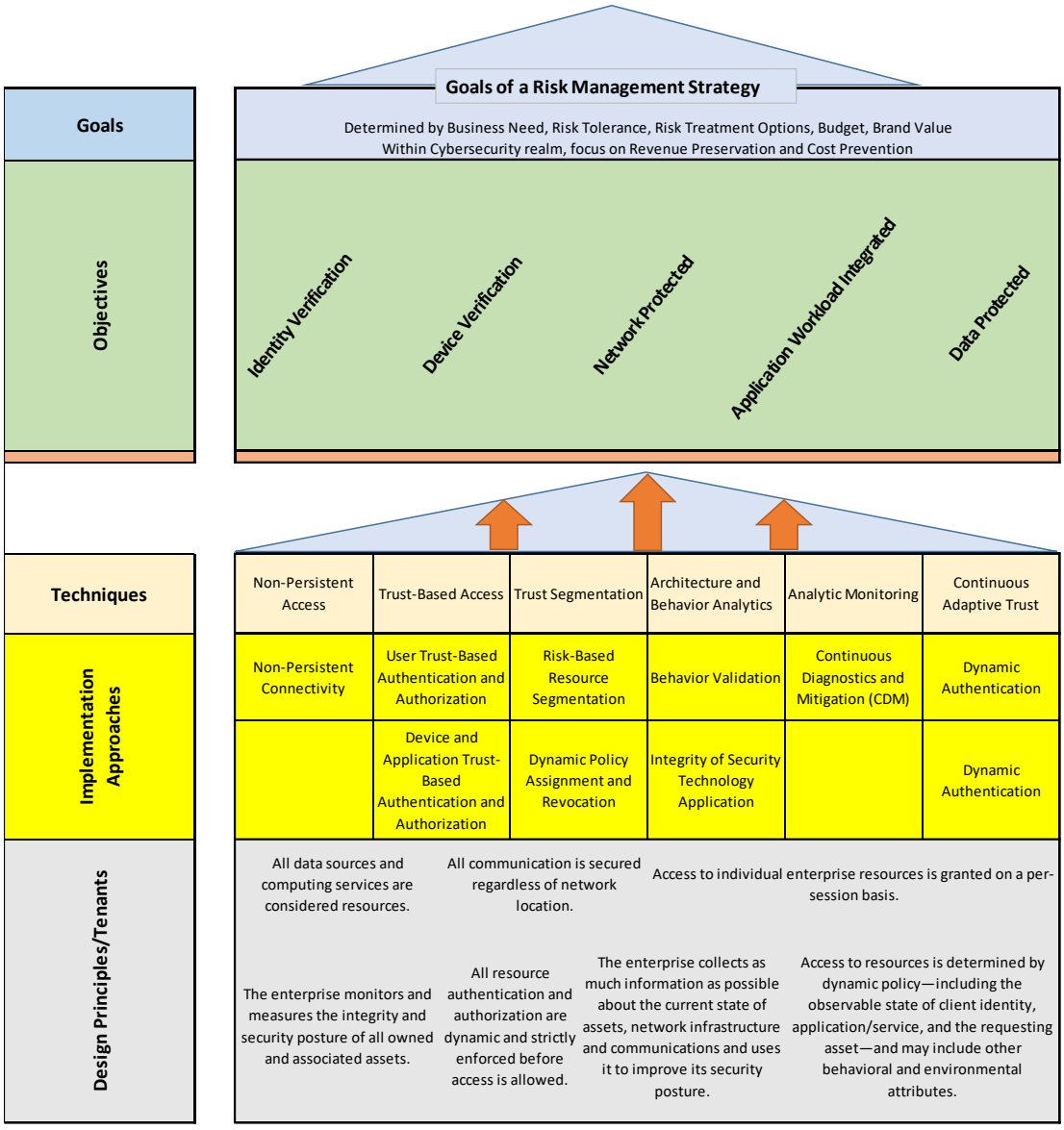


Figure 1: Applying CISA Cyber Security Model with NIST Cyber Resiliency Constructs to Implement Zero Trust Architecture

Trust and Decision Architecture

Zero trust architecture uses identity and the context around the identity of a subject to manage adaptive trust decisions. Zero trust is all about creating point-in-time explicit trust for certain access at a certain moment and in doing so, it must have the tools and data available immediately for the policy administrator to determine a subject claim of identity. And since a huge part of subjects (anything trying to access the resources of another thing) is humans, a robust identity access foundation is critical for success [5]. The "source of truth" of John Doe actually being "John Doe" and various attributes about him being accurate and available at all times to help the policy administrator is the heart of zero trust and the first pillar of our Objectives for our custom Zero Trust Framework from Figure 1. Later in this paper we will operate a case study for John Doe from the Design Principles all the way up through the Identity Objective to accomplish a Risk Management Strategy goal.

Zero trust architecture uses existing tools and technologies in concert with each other for them all to contribute to the overall protection of the enterprise. Some of these tools might sound familiar in an implicit-trust, perimeter-based network of today. Use some firewalls and some IPS/IDS at the perimeter and then some EDR tools on endpoints and ultimately send all logs to a SIEM for analysis and alerting? Maybe implement some IPSec tunneling and even use mutually authenticated TLS (Transport Layer Security) for internal encryption flows and enable encryption of data at rest on the SAN? All of these technology implementations are good ideas and very necessary and valuable to any given network and are still relevant during and after implementation of zero trust. Where the technical implementation and sometimes novel use of security technology is differentiated in a zero trust network is the existence, reliance, and importance of something called the control plane and the data plane. The control plane is the heart and soul of zero trust architecture and this is where the information gathered and supplied by all other cybersecurity tools, alongside Identity Management tools, Asset information, configuration information, SIEM, Continuous Diagnostics and Monitoring tools, and many others are

collected and contribute to the policies that ultimately are executed on the decisions based on this information. The control plane will likely become a Crown Jewel of the organization and will require the utmost attention and protection, as these are the logical components of an on-premise or cloud-based service which operate independently outside of the traditional data flow network (that segregation is tremendously important) yet makes all of the decisions regarding access to resources in that data flow network. The control plane includes two critical features of zero trust architecture: The policy engine (PE) and the policy administrator (PA). Collectively, these two are referenced as the policy decision point (PDP), which as the name describes, makes all the decisions about if and how network resources are going to connect to each other and if they are going to be successful in doing so. Once the decision is made by the policy administrator, that decision is passed to the data plane (which resides on that aforementioned data flow network) where the Policy Enforcement Point (PEP) enables the freshly-established trusted relationship between the subject and the resource, monitors it, and eventually terminates it per the request sent from the policy administrator (See Figure 2) [1].

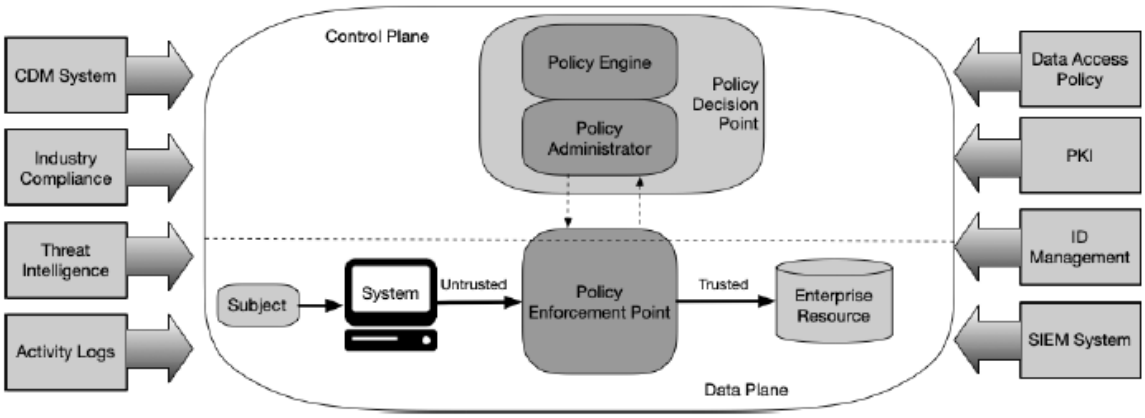


Figure 2: Core Zero Trust Logical Components [NIST SP 800-207]

What does the policy engine do with all of this ingested information and data about subjects in order to feed the policy administrator what it needs to make a decision on granting or denying access to a resource? It uses a Trust Algorithm (TA). The Trust Algorithm I like to think of as the calculation of a

trust score. Various inputs of information are fed to the PE and each one holds a certain value. Does the user exist? Did they provide MFA credentials in their request? Are they requesting a connection from a known corporate device? Was the connection initiated via an encrypted tunnel originating from a TPM-enabled device? Is it normal work time? Is it the same time as this user and device have tried to access this resource before? All of these and many more questions ultimately lead the PE to calculate a certain "trust score" for the connection and if a certain threshold of a trust score is met as per required for the resource, the PA allows the PEP to create the connection. Certain resources may require different trust scores in order to gain access, just as certain trust scores of subjects (maybe a new or transferred employee), may prompt for additional input in order to raise their score high enough to access a resource. This all occurs very dynamically and the PE must be executing those various trust algorithms against every request that flows through the network.

Risks

Due to the complexity and collaborative nature of disparate systems communicating extensively in real time to make expedient decisions on access and privilege based on a multitude of factors from every user, device, and resource on the network, there is very high risk to impacting business operations during and after implementation of zero trust. False positives could inundate the systems' monitors and alerts and reflect a failure in the architecture, but more impactfully it could prevent an authorized user from retrieving needed data to run the business. In a zero trust network, there are going to be possibly thousands of opportunities to block valid users and resources from performing valid business functions. Since the purpose of cybersecurity is to support the business and contribute to the bottom line either in the form of revenue protection or cost prevention, implementing a cybersecurity solution which negatively impacts the business operations is just not acceptable. Additionally, as mentioned before in the many thousands, or possibly millions, of Trust Algorithms which must be processed by the policy engine and decided upon by the policy administrator in the control plane and then sent to the Policy

Enforcement Point to establish and terminate the connection, a bottleneck can come to fruition quite quickly and bring the data transport at the organization to its' knees. Having to make a data-informed decision automatically and then execute on blocking or creating the connection at the speed of business is not only at risk when it fails, but also at risk when it does not scale at the speed of the demand.

Zero trust architecture is also not a magic shield of protection from every malicious activity possible. An unaware accountant might still click on a phishing link and deploy malware into the network. Zero trust will likely contain the blast radius of the malware and prevent it at various steps of the cyber kill chain, but malware infection can still happen. Supply chain compromises of hardware, depending on the status of the TPM chip (Trusted Platform Module) and hypervisor vulnerabilities can undermine the separation of the data and control planes, where policy is going to be centralized and enforced. Encryption technologies required for zero trust implementations may make monitoring and anomaly detection more complex, and deep packet inspection impossible. A compromised system may be able to use that same encryption to hide malicious activity from detection. Legacy applications and even organizational resistance can prevent implementation of the required components of zero trust and even create the vulnerability necessary to exploit the network and "hide" from the very protections zero trust is meant to enable. Additionally, zero trust is protecting all of the individual resources on the enterprise network, so it does not come into effect when the organization is hit with a DDos attack. Zero trust focuses on authentication and authorization, and thus a high volume connection-request attack such as DDos will still require additional focused protections outside the control plane and data plane of zero trust.

Deeper Technical Requirements and Recommendations

Zero trust architecture requires that all communication requests are expected and data is protected by the best encryption possible, at rest and in transit. Transit can be tricky, as there must be a fully established trust chain created to ensure that the origin source of the data, the transport medium,

and the recipient of the data all are trusted that no gaps were left unclosed in their communication and inadvertently affect the confidentiality or integrity of the data. X.509 certificates are a popular and preferred choice for authenticating TLS (Transport Layer Security) connections. Mutually authenticated TLS is the recommended communication protocol for internal client/server connections within the enterprise [2], as it is supported by most Application Layer protocols. Side note, although complex and potentially expensive, for true transport defense in depth a company can use mutually authenticated TLS in addition to IPsec (Layer 7 in addition to Layer 3) [2] but to save on costs it is recommended to use IPSec in transport mode for server to server interactions, or UDP encapsulation for networks which do not support IPSec. Certificates like X.509 use a public and a private key, where the public key is distributed and the private key is held as a secret. The private key is required to decrypt a cipher that was encrypted by the public key. This ensures that the data is not decrypted by anyone except the holder of the private key, but a private key is "something you have" authentication…and anything you have can be stolen. This can be mitigated by practices such as credential rotation or the usage of multiple secrets and it can have more protection against theft by using privately-signed Certificates from the organizations own Certificate of Authority (CA) rather than a public CA. When used in conjunction with TPM chips, X.509 certificates with strong management practices are the most robust security of our data origination and destination [2]. Additionally, implement and use least-privilege administration and configuration monitoring, such as Linux-based solutions like Ansible, Puppet, Chef and Microsoft Desired State Configuration (DSC) and PowerShell concepts of Just In Time (JIT) Administration and Just Enough Administration (JEA) [6].

Conclusion

Zero trust architecture is a system of principles and decisions and tools and planning that all ultimately work together to underscore some fundamental assertions [2]:

- The network is always hostile.

- External and Internal threats exist at all times and all around every system.
- The historical paradigm of location within the network determining implicit trust is not enough.
- Every user, asset, flow, resource subject will be authenticated and authorized.
- Policies must be dynamic and ingest as many sources of data as possible to make trust decisions.

As we have discussed throughout this paper, zero trust is essentially a security paradigm for making sure that people and entities attempting to connect to company resources are who they say they are, which requires explicit authorization for every request after a comprehensive authentication exercise and continuous monitoring of all activity to look for signs of unexpected activity. This goes far beyond basic authentication of the old days when a username and password in Active Directory could get someone access to any system on the network. This type of access management assumes that all users are a threat, regardless of their identity, location or how they connect to a network (be it "inside" a company network perimeter or remotely). It is important to note that zero trust is an evolution, not a revolution [7]. William Malik, vice president of infrastructure strategies at Trend Micro, stated that "Don't try and buy your way to zero trust – set small goals, make sure it is rooted to removing un-earned trust, and always ensure that you have visibility improvements" [7]. This is why my mind went to how to build a framework and approach mechanism in this paper in order to create focus and priority in the efforts that a company can use to build towards a zero trust future. Use the principles to make a plan to implement, see what people and technology you need, measure along the objectives, and prioritize progress to the goals. Building an architecture that never trusts and always verifies to grant access explicitly and intentionally on a per-request basis and assumes every system is surrounded by adversaries and a threat actor is active at all times leads to highly resilient, highly flexible environments that are much better suited to the demands of the modern workplace of today and to prepare for the threats of tomorrow [5].

Creating a Zero Trust Architecture Framework

Introduction

In this section, I will present an application of the proposed framework which was referenced in this research paper. I will include the definition of the Design Principles of zero trust, establish Implementation Approaches, propose Techniques, and link them all to Objectives which will ultimately support the Risk Management Strategy goals of an organization. By combining and defining zero trust architecture principles with NIST cyber resiliency constructs and CISA maturity model assessment, I will create a familiar, yet unique, guide for an organization to determine their logical path towards a zero trust architecture.

Design Principles/Tenants

According to NIST SP 800-160v2, developing cyber resilient systems is founded in defining structural and strategic design principles. These are the initial underlying principles which engineers and architects can use to guide and inform design decisions and analysis [3]. From these principles, various constructs build upon each other to ultimately achieve one or more of the four goals of resiliency: Anticipate, Withstand, Recover, and Adapt. Zero trust architecture is not defined in the same way by everyone who designs a system or does zero trust must have an agreed upon mechanism for measuring "success". Even beginning an initiative to embark on the journey of zero trust can come with varying recommendations of where to begin. One authors' recommendation might be to catalog all data flows in the network while another might be to implement a centralized Identity and Access Management (IAM) System. How a company and the architects and engineers inherently operate will greatly influence what recommendations they follow, usually playing into a set of core competencies. If the architect is skilled in inventory analysis then they may begin with data collection of assets and data flows. If the engineer is very technically focused on implementing tools and knows that a centralized IAM would be necessary for zero trust or thinks that full encryption of data at rest and in transit is part of zero trust,

they may begin with those types of technical implementations. Throughout my research of reading books, articles, interviewing engineers who have "implemented" zero trust both as a professional service provider to a customer and with Microsoft engineers who implemented their version of zero trust for Microsoft's own network, I have come to the conclusion and recommendation that the tenants of zero trust as defined in NIST SP 800-207 [1] have a direct correlation to act as structural design principles to underset building a framework using constructs as in NIST SP 800-160v2. Using these tenants as design principles has allowed me to systematically build up through the various processes and technologies that would be required to ultimately achieve the Risk Management Strategy goals of an organization. See figure 3 for how the zero trust tenants of SP 800-207 correlate to cyber resiliency structural design principles of SP 800-160v2.

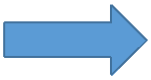
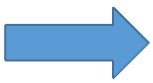
ZTA Tenants		Cyber Resiliency Structural Design Principles
All data sources and computing services are considered resources.		Limit the need for trust.
All communication is secured regardless of network location.		Make resources location-versatile.
Access to individual enterprise resources is granted on a per-session basis.		Contain and exclude behaviors.
Access to resources is determined by dynamic policy—including the observable state of client identity, application/service, and the requesting asset—and may include other behavioral and environmental attributes.		Plan and manage diversity.
The enterprise monitors and measures the integrity and security posture of all owned and associated assets.		Control visibility and use.
All resource authentication and authorization are dynamic and strictly enforced before access is allowed.		Determine ongoing trustworthiness.
The enterprise collects as much information as possible about the current state of assets, network infrastructure and communications and uses it to improve its security posture.		Make the effects of deception and unpredictability user-transparent.

Figure 3: NIST Zero Trust Tenants Mapped to NIST Structural Design Principles

Implementation Approaches and Techniques

Depending on the prioritization of the certain objectives to meet the goals of zero trust architecture, the design principles inform the selection and prioritization of what implementation approaches to focus resources on and the techniques that ultimately will support the proper objectives. Since most organizations will need to use a phased migration approach to zero trust and thus maintain positive user experience and minimize employee frustration, choosing the implementation approaches and techniques to best balance the migration to greater security yet minimizing impact to business operations is critical. Every organization has their own interruption tolerance level and end user technical aptitude, and these factors will also come into heavy consideration when selecting the implementation approaches and techniques selected to move towards the goals defined by the organization.

Correlation directly to cyber resiliency constructs of implementation approaches and techniques is not as streamlined as observed for design principles, but in definition of these constructs and the macro-level view of zero trust made creating and defining these for zero trust relatively straightforward. The technology around zero trust is not necessarily “new”, but it is the systematic application of all of the concepts working in unison that makes zero trust so unique. Many organizations have used and practiced parts of the recommendations of zero trust for many years, so creating the implementation approaches and techniques which would connect the design principles to the objectives is really only a matter of bringing all of those parts together into creating a whole comprehensive solution.

Objectives

In designing a cyber resilient system, the objectives are where the progress and measurement to the goals are tracked and measured. This is a critical construct area as it is here that the organization must determine the correct balance of risk tolerance vs diminishing returns on investment in addition to calculating the security investment value to risk mitigation. In this construct the organization measures the progress to the Risk Management Strategy goals and, even at a basic level, must calculate if the

investment in zero trust tools and processes is accomplishing the basic cybersecurity value-propositions of cost reductions or revenue preservation. With no universal definition of zero trust objectives, it would be left to the evaluation of the organization itself. In my research, matching seamlessly to the definition of the "objectives" construct are the pillars of the CISA Zero Trust Maturity Model [4]. Every technical and operational control I have researched as pertains to a zero trust implementation could be categorized back to one of these maturity-model pillars as shown in figure 4 and application of various techniques and approaches to the Objectives in figure 5.

Pillars Redefined into Objectives:	Description:
Identity Verification	Multiple factors, such as behaviors, tokens, biometrics, certificates, PIN, passwords, BYOI, etc. applied dynamically to verify authenticity of claim of Identity by a subject requesting access to a resource
Device Verification	Multiple attributes, such as TPM chip, certificate, location, configuration, installed software, etc. supplied dynamically in support of a user authentication request for access to a resource
Network Protected	Network acts as an inclusionary vessel for secure data transport and responsible for bootstrapping trust via pre-authentication and ensuring all network flows are authenticated before being processed. Network does not act as a perimeter but ensures all traffic is encrypted and uses ML to inform threat protection mechanisms
Application Workload Integrated	Integrate threat protections and application health processes into application workload. Applications respond instantly to Policy Enforcement Point access connections and terminations.
Data Protected	Data is encrypted with the strongest encryption possible in transit and at rest. Data is identified, catagorized, and tiered appropriately in order to be available for authorized subjects providing the appropriate trust score.
<<<--- Visibility and Analytics	Automation and Orchestration Governanace --->>>

Figure 4: CISA Maturity Model Pillars Redefined into Zero Trust Objectives

Techniques/Approaches \ Objectives	Identity Verification	Device Verification	Network Protected	Application Workload Integrated	Data Protected
Non-Persistent Access	X		X	X	
Non-Persistent Connectivity	X		X	X	
Trust-Based Access	X	X			X
User Trust-Based Authentication and Authorization	X				X
Device and Application Trust-Based Authentication and Authorization		X			X
Trust Segmentation			X	X	X
Risk-Based Resource Segmentation			X	X	
Dynamic Policy Assignment and Revocation				X	X
Architecture and Behavior Analytics	X	X	X		X
Behavior Validation	X	X	X		
Integrity of Security Technology Application			X		X
Analytic Monitoring	X	X	X		X
Continuous Diagnostics and Mitigation (CDM)	X	X	X		X
Continuous Adaptive Trust	X	X			X
Dynamic Authentication	X	X			
Dynamic Access and Privileges	X	X			X

Figure 5: Mapping of Zero Trust Objectives to Proposed Techniques and Approaches

Goals

What is the goal of zero trust? Some might say that it is to protect organizations from threats and to protect the data that might be valuable to someone who is not supposed to have it, either to steal or to exploit the company. An organization should not design their goals around zero trust by checking boxes of technical accomplishment of what any vendor or article or periodical says is the definition of “Zero Trust”. The organization needs to incrementally implement zero trust principles, process changes, and technology solutions that protect its highest value data assets [1] and in order to

do so, they must have a clearly defined Risk Management Strategy. The organization must know what their valuable assets are in order to know which ones have the highest return on protecting. If the company is attempting to protect everything in the organization, there is inevitably a point of diminishing returns on that level of investment and an opportunity cost of resource focus away from the most valuable assets and data to the org. Cybersecurity exists in order to manage risk, and the principles of zero trust provide the guidance to apply capabilities to minimize that risk in such a comprehensive manner it has not been seen before in this industry, but it not a magic bullet that will save the company from everything and there is a treatment of risk that is very much applicable to any conversation because to accept it is a much better business decision than the investment to mitigate it.

Implementing Zero Trust Architecture

Introduction

In this section, I will propose how an organization can take steps toward zero trust architecture categorized by the objectives defined in the framework in the previous section. To conclude, I will test this implementation against the "Identity Verification" Objective and measure how John Doe will access various resources within an organization and how the framework ensures that the assertions of zero trust architecture are applied.

Identity Verification

I recently went on a business trip to Redmond, Washington to meet with Microsoft engineers and got the rare opportunity to talk with a cybersecurity architect and ask about how Microsoft implements protections of and from their 500,000 pc's, tablets, iPads, laptops, and mobile phones and the people using them. What I observed was textbook best practices in some regards of implementing the principles of zero trust and support of my defined objective of Identity Verification. Utilizing heavy use of MFA, this particular architect was in possession of no less than four different smart cards used as

the “something you have” component of MFA and each card had access to a varying level of system. In addition to the authorization level provided by the presentation of the smart card, Microsoft utilizes Microsoft Hello biometric logins and certificate-exchange PIN logins. They are exclusively passwordless, demonstrating that adhering to the facets of MFA does not inherently assume one of those factors must be a password. Using Conditional Access Policy, which also applies to the device they are using (discussed in next section) the trust established for a users’ authentication and authorization is an ingestion of a multitude of signals which ultimately allow for the policy enforcement to execute or block the access request. The combination of these signal inputs and the conditional access policy can also allow for machine learning to adopt user behaviors and apply those to the risk signals [8] incorporated into the policy engines. For example, if a user has logged in on a certain device from a certain location and only during working hours for a period of time and suddenly there is a request from that user on a weekend from a location a thousand miles away, the policy has the knowledge to block the connection or require a deeper level of authentication.

“Identity is the new perimeter” was my first easily understood definition of zero trust. This tenant holds true but accomplishing zero trust takes more than just wrapping static security around identity, it takes a level of dynamics to ensure that an identity is always challenged so that a compromised identity could not proliferate very substantially. Credentials should ideally rotate between every session so that an adversary cannot reuse credentials. It is critical that credentials be time-boxed to minimize the blast radius of leaked or stolen keys and hashes and gives the subject an opportunity to continuously reassert trust [2], which could also increase trust scores for some users enabling even more efficient access in the future. Speaking of efficient access, Single Sign-On (SSO) is a common efficiency mechanism implemented to pass-through credentials between systems and networks, essentially passing implied trust from one system to the next. There is a centralized authority granting a token which validates the subject and passes the authentication through and this is in conflict with the

zero trust principle of decentralized authentication. In zero trust, it is the control plane that makes the decisions on authentication and pushes the dynamic credential and access policy to the data plane [2]. In order to implement zero trust and constant authentication, but still receive the benefits of SSO, the token must be passed through the control plane and re-authorized for every access request between systems rather than simply passed from one system to the next system.

MFA is on the verge of a disruption that fits seamlessly into the future of zero trust, and that disruption is what Gartner is naming Continuous Adaptive Trust (CAT). This evolution is not about two or more factors provided that identify a user (MFA), but how credentials are combined with recognition, affirmation, and risk signals to provide sufficient trust in an identity claim [9]. Claimed identity and access risk can change dynamically throughout a session, so credentials and signals must be continuously evaluated postlogin [9] and this is essential to maximize the protections afforded by implementing zero trust. This CAT concept matches seamlessly with what Microsoft is doing with their Conditional Access Policy. MFA is not required for every login, but can be invoked if necessary, as the policy engine is ingesting and evaluating every identity, device, and risk signal that is being supplied by the connection and running analytics on user behavior, session attributes, familiarity attributes, and threat intelligence, and environmental context [9].

Device Verification

At the heart of every device, and a strategy that should be implemented in every organization of every size, is the embedded TPM (Trusted Platform Module) chip on every server hardware and every end user computing device. The presence of this cryptoprocessor is critical for device authentication in a zero trust network. Why are they so critical? They allow for cryptographic operations yet have no interface for retrieving private keys. So, when a TPM generates and stores a Storage Root Key (SRK) and shares the public key, it can only be decrypted by the private key stored in that originating hardware TPM. Asymmetric encryption is expensive and slower than symmetric, so the device will use a fast

symmetric encryption such as AES to encrypt the bulk data, and then use the SRK to encrypt the resulting AES key [2]. Additionally, authentication on any device requires the use of x.509 certificates and the most secure storage for a particular device's X.509 private key is within the storage backed by the TPM. The marriage of TPM and X.509 certificates is foundational and critical to devices in a zero trust environment.

Devices also contain attributes that must contribute information to the policy engine in a zero trust network. Configuration of the device, applications installed on the device, physical location of the device and how it is connecting to the Internet are all considerations which contribute to the device getting authorization to connect to data sources. An organization can streamline management and capture of these attributes with an implementation of an MDM (Mobile Device Management) or Endpoint Management tool such as Microsoft Intune or Citrix Endpoint Management. Combining the management of those devices with a Conditional Access Policy from Microsoft will allow for the granular control of associating the static physical attributes of the device with dynamic point-in-time environment attributes of the device to contribute to the policy administrator decisions to allow access to certain data sources. For example, an employee laptop connected through a connected home internet connection that meets all patching and configuration criteria will present a higher trust score when requesting access to a company data source than that same laptop connecting over a free Wi-Fi connection from a hotel lobby in a foreign country.

Network Protected

In the early days of cybersecurity, it was thought that a company needed to protect the "network" to protect the company from cyberthreats. An evolution in the capabilities and motivations of adversaries has taken the fight inside the network. This evolution has created the necessity for a maturity of understanding what entails a network, what is "inside" that network, and how that network is going to communicate. Networking technology applied correctly is pivotal to any zero trust

architecture. Gone are the days of a single-segment corporate perimeter being sufficient and introduced in zero trust is the implementation of microperimeters around the most granular of systems and microsegmentation of discrete zones based on sensitivity of data or functions of the business. The most valuable data in the world is useless if it cannot be moved, and the network is the catalyst to adding value to data to get to the proper destination securely and begin transformation of data into information and knowledge. The Network Protected objective is accomplished by implementing a collective of existing technology, but in a novel way. Firewalls are still required, albeit they may separate users from applications and applications from databases (east/west traffic) rather than just the traditional corporate-from-DMZ-from-Internet (north/south). Even host-based firewalls directly on the system should be implemented and enabled to further supplement any network firewalls. Many periodicals suggest that mapping of network flows is the first step to take in implementing a zero trust network, and although this statement may be accurate for some organizations depending on what they have determined is their best implementation approach (Behavior Validation in the above framework), there are many additional technical considerations to achieving this objective.

One of the most difficult concepts of zero trust to grasp (for me, at least) in how it would work in operation is the concept that VPN is no longer necessary or valid if the organization has implemented zero trust. VPN typically uses IPSec encrypted communication in tunnel mode establishing a secure connection between two endpoints over the Internet. The termination point of an endpoint is a zone of implied trust, thus negating one of the main tenets of zero trust. Using that existing technology, but in transport mode and in higher volume so that the link goes directly to the destination endpoint rather than an intermediary network device (i.e., firewall), creates a secure connection between the subject and the data source thus rendering the location of either entity obsolete. Traditional VPN in tunnel mode encapsulates the entire IP packet, but with more IPSec connections going directly to the endpoint and creating more volume of data transfer and encryption/decryption overhead, using transport mode

allows for only the payload to be encrypted, which still ensures that security is enforced from end to end.

With network location now irrelevant, zero trust architecture requires we look more closely at the communication and mechanisms needed for implementation and accomplishing the objective. IPSec is recommended for server-to-server communications within the data center, as it relieves some of the additional overhead and complexities that come with the recommendation of client-to-server communication, mutually authenticated TLS. IPSec is not a single protocol, but a collection of protocols and one of those protocols is the Internet Key Exchange (IKE) which is the protocol that performs the authentication and key exchange components of IPSec [2]. Think of IKE as the control plane of IPsec and this is where the X.509 certificate, again, is used to authenticate a peer and authorize a connection. In a zero trust network, all devices should be using x.509 certificates issued ideally from a private Certificate Authority (CA) that is offline rather than a public PKI (Public Key Infrastructure). X.509 certificates are meant for all devices in the environment, not humans, as they carry proof of trust, signed metadata, and a way to strongly encrypt data using its identity [2].

Client-to-server connections typically will run applications which use protocols that support TLS (Transport layer Security) running at application layer 6 of the OSI model, whereas IPSec generally runs at layer 3 or 4. This puts the encryption at the application rather than the session and is evident in common web services using https and secure email, for example. In TLS, only the data source is authenticated, but the client is not. This is why anyone on the Internet can access a website using https…it verifies the destination, but the origin can be anyone. Although the communication to the destination would be encrypted and protected with X.509 certificates, zero trust principles mandate that the client is authenticated and authorized. To accomplish this, the handshake of a TLS connection will verify and authenticate the client requesting the connection. This type of connection is referred to as “mutually authenticated” TLS, or mTLS and is a requirement to conform to the zero trust model [2].

Securing the network and the connections is covered and authenticating any device or user requesting access to the network is covered, but both of these concepts have an assumption that that connection did not meet interference or compromise while it was being created. So how does the connection get created in a zero trust network before any of the above activity can take place? That answer lies in Single Packet Authorization (SPA) and bootstrapping the trust of the proposed incoming connection. SPA is a type of pre-authentication technology that works by sending a small piece of encrypted data to a destination to set the expectation of the incoming secure TLS or SSH connection. A common implementation is to use fwknop (firewall knock operator) which works by having a daemon listen to UDP port 66201 on a firewall [10] (or reconfigured via command line) and when the packet is received, decrypted, and inspected the payload includes the protocol and port numbers the subject is requesting access to [2] which then creates the firewall rules permitting the requested connection. For additional security, fkwnop can be configured to add an HMAC (Hashed Message Authentication Code) to the end of its payload, which prevents tampering by guaranteeing that the message is authentic [2].

Application Workload Integrated

Every application in the organization must be protected as if it is directly connected to a very hostile Internet, because it is. Legacy applications must receive connections via an application proxy which provides the same protections and access justifications as modern applications built with zero trust principles in mind. COTS (commercial off the shelf) and in-house developed applications must be able to respond immediately to access executions and revocations the instant they are relayed the proper information from the Policy Execution Point and do so in a secure manner utilizing valid X.509 certificates.

Application developers must ensure data integrity throughout the systems development lifecycle by employing a secure repository which follows least access privilege principles. In a version control system such as “Git” cryptographic hashes of ancestor commits must build on new commits to

form a Merkle Tree, which allows for cryptographically validated assurance that the chain of commits is unmodified [2] and signed with the GPG (GNU Privacy Guard) key of a trusted developer. The development platform, such as Azure DevOps, must be connected to the build server over a secure TLS channel and the build server should confirm all signatures before starting a build. Artifacts generated by the build server should have immutable properties (write once, read many) before producing later versions for distribution. The home source of truth for all developed applications must be afforded the highest levels of protection.

API integrations are a seamless way for applications and services to interact and create valuable access to information for organizations. A single organization can have hundreds of APIs to allow for disparate applications to communicate and share data, but they can also be unprotected threat surfaces prime for attack by an adversary. APIs need to be protected by least privilege principles and strong identity management combined with micro-segmentation [11]. In an age of continuous integration/continuous delivery (CI/CD) and the criticality of APIs, adding additional layers of security can be an afterthought or a bolt-on. Implementing zero trust principles at the endpoints where APIs are delivered at every stage of the SDLC and through promotion and connection to other APIs (which communicate using the Network Protected implementation standards noted above) is critical to the continued benefit of dynamic data integration in systems.

Data Protected

Almost every implementation approach and technique has a connection to the “data protected” objective. Is this surprising? No, it is not surprising as data is at the core of what an organization is trying to protect with any implementation of cybersecurity processes and tools. It is not necessarily the physical server or the tablet or even the operating systems and applications that hold value to the organization, but it is the data that those tools provide access to that requires protection. Data is the target of an adversary’s goals of deny, deceive, disrupt, deter, or destroy because data is what creates

information which converts to knowledge and knowledge is what is needed to run an organization and knowledge transfer created into a tangible or intangible product is how the organization makes money.

In order to protect data in a zero trust architecture, systems must be employed that can provide services no number of human eyes and hands could ever perform. Zero trust requires continuous monitoring of all systems in the environment. All activity must be baselined for normalization trending and configured to alert to anomalies. Advances in machine learning and artificial intelligence must be utilized in ways to inform rapid information gathering and supplying to the control plane, policy engine and trust engine. All authentication, transport, ingress, egress, and access activity on the network must be logged and sent to a centralized SIEM for deeper analysis and alerting. Threat hunting feeds and configuration vulnerability reports must be sent to the control plane in addition to the logs and information from the SIEM. The control plane must be protected and treated as one of, if not the, most critical systems in the environment since it acts as the gatekeeper to accessing every bit of critical data within the organization.

Part of protecting data using least access privileges also incorporates change management. A Request for Change (RFC) is a known security and configuration management best practice widely adopted to enterprises worldwide. To maximize protection of data and elevate the maturity of the data protection objective an organizations' policy administrator should allow access authorizations to be granted only during approved change windows. Access should be immediately revoked at the termination of the change window and configuration of the changed system scanned and updated to the Configuration Management Database (CMDB).

Test Case: John Doe

In this test case, John Doe is an employee of Rhino Analytics. He is working remotely from his home using an employer-provided tablet and connecting to the corporate email and finance system. The email system is hosted in Exchange Online and the finance system is hosted in the corporate data

center. Using the framework above, I will demonstrate how John accesses these two company resources via zero trust architecture and the following NIST Zero Trust Architecture tenants and my proposed framework structural design principles:

- All data sources and computing services are considered resources
- All communication is secured regardless of network location
- Access to individual enterprise resources is granted on a per-session basis
- Access to resources is determined by dynamic policy—including the observable state of client identity, application/service, and the requesting asset—and may include other behavioral and environmental attributes
- The enterprise monitors and measures the integrity and security posture of all owned and associated assets
- All resource authentication and authorization are dynamic and strictly enforced before access is allowed
- The enterprise collects as much information as possible about the current state of assets, network infrastructure and communications and uses it to improve its security posture

John's zero trust journey to access the company resources he needs begins immediately when he powers up his company-owned tablet. That tablet contains an embedded TPM cryptoprocessor which ensures that all activity on that device has maintained full integrity through the supply chain cycle of construction and delivery and stores the private key to decrypt the encrypted key of bulk AES encryption and the X.509 device private key. John attempts to log onto his tablet first by plugging in a YubiKey plug-in hardware authenticator device and providing the facial recognition prompt sent to his cell phone by FIDO2 [https://fidoalliance.org/fido2] certified SaaS authentication provider, Hypr.

As the tablet loads the Operating System and John's profile, the Crowdstrike Falcon XDR, Splunk SIEM, and Microsoft Intune agents load, establishing their secure SSH transport-mode encrypted connections to their host services. John's login activity and all activity he performs is now being monitored and logged and sent to Rhino Analytics' control plane for analysis by the policy engine, trust engine, and policy administrator. First, John checks his email by initiating a secure TLS connection to the SMTP servers of Microsoft Exchange Online. The Microsoft Conditional Access Policy evaluates John's location and the configuration of his tablet to ensure the latest updates are installed and he is an

authorized geographic location before allowing the connection to the destination to retrieve his email. As John is reading his email, he receives a file attachment from a peer. He opens this file and the file opens in a sandboxed secure area of Microsoft's CASB (Cloud Access Security Broker) to evaluate the contents of the file to look for malicious code. Upon delivery of the contents of the file to John, his system is performing another inspection on the contents of the file and the company control plane initiates another verification of the configuration of John's tablet and of John's authentication credentials to ensure these have not changed. These have not changed so John's trust score has increased by one.

Now that John has completed reading his emails, he now needs to access sensitive company financial data serviced from an on-premise server in the data center at Rhino Analytics headquarters. John opens his client application and the information is sent to the policy engine and the policy administrator determines that John needs additional authentication in order to access the financial data source. After providing a certificate-backed PIN number which provides the private key to the destination applications-presented public key, the control plane sends a request to the financial application for verification of its TPM module encryption status and x.509 certificate and once those are authenticated, verifies the level access John is authorized for. Once the control plane has gathered all of this information from both parties and determined that both identities match their claims and that John's request meets his level of authorization, the control plane informs the data plane to establish a mutually authenticated TLS connection between John's tablet at home and the company on-premise financial application. Intermediary filtering by corporate perimeter firewalls and host firewalls on both John's client and the hosting server perform constant analysis and packet inspection (minus payload, since that is encrypted) and continuously feed information which ultimately services the policy engine. John completes his work in the financial application and terminates the connection and the control plane again initiates a review of the configuration of John's system and the financial system to look for

changes, and John's temporary credentials used for his access is destroyed so they can be re-created again upon his next access request.

All of this secure activity was invisible to John and he was not hindered in any way to perform his job functions. John is who he claims to be and is authorized to access all of the systems and data he attempted to access, so his experience was streamlined and that is going to be critical to avoid causing the business friction and create user defiance. To enable as seamless as possible user experience, implementing zero trust is a journey and must be done iteratively to ensure the systems are built to keep up with all the demand and execution on the decisions in near real-time in addition to not preventing employees from performing the work they are expected to perform. In this simplified example I was able to define how achieving every zero trust tenant was successful and how it significantly protected the organization from any resource trying to perform any activity other than those authorized and by anyone other than who they had to prove they are.

References

[1] S. Rose, O. Borchert, S. Mitchell and S. Connelly, "Zero Trust Architecture: NIST SP 800-207," August 2020. [Online]. Available: https://doi.org/10.6028/NIST.SP.800-207. [Accessed 24 June 2022].

[2] G. Evan and D. Barth, Zero Trust Networks: Building Secure Systems in Untrusted Networks, Beijing, Boston, Frnham, Sebastopol, Tokyo: O'Reilly, 2017.

[3] NIST, "Developing Cyber Resilient Systems A Systems Security Engineering Approach SP800-160," NIST, November 2019. [Online]. Available: https://doi.org/10.6028/NIST.SP.800-160v2. [Accessed 14 March 2022].

[4] CISA, "Zero Trust Maturity Model," June 2021. [Online]. Available: https://www.cisa.gov/sites/default/files/publications/CISA%20Zero%20Trust%20Maturity%20Model_Draft.pdf. [Accessed 15 June 2022].

[5] C. Winckless and N. MacDonald, "Quick Answer: How to Explain Zero Trust Technology to Executives," 27 September 2021. [Online]. Available: www.gartner.com. [Accessed 20 June 2022].

[6] P. Ferrill, "What Is Zero Trust Architecture?," The New Stack, 17 June 2022. [Online]. Available: https://thenewstack.io/what-is-zero-trust-architecture/. [Accessed 24 June 2022].

[7] T. Seals, "Zero-Trust For All: A Practical Guide," February 2022. [Online]. Available: www.threatpost.com. [Accessed 11 June 2022].

[8] Microsoft Docs, "What is Conditional Access?," Microsoft, 22 April 2022. [Online]. Available: https://docs.microsoft.com/en-us/azure/active-directory/conditional-access/overview. [Accessed 28 July 2022].

[9] A. Allan, "Shift Focus From MFA to Continous Adaptive Trust," 01 December 2021. [Online]. Available: www.gartner.com. [Accessed 07 July 2022].

[10] Matthias, "Using 'fwknop' on OpenWRT," 15 March 2015. [Online]. Available: https://matthiasl.github.io/output/Using__fwknop__on_OpenWRT.html. [Accessed 31 July 2022].

[11] L. Columbus, "Why the future of APIs must include zero trust," VentureBeat, 01 August 2022. [Online]. Available: https://venturebeat.com/2022/08/01/why-the-future-of-apis-must-include-zero-trust/. [Accessed 01 August 2022].

[12] C. Winckless and S. Olyaei, "How to Decipher Zero Trust for Your Business," 22 May 2022. [Online]. Available: www.gartner.com. [Accessed 17 July 2022].

[13] M. James, "A 2022 Guide to Zero Trust for Data Protection," Smart Data Collective, 2022. [Online]. Available: https://www.smartdatacollective.com/guide-to-zero-trust-for-data-protection/. [Accessed 28 July 2022].