Zero Trust Security

A Complete Guide

NIKE ANDRAVOUS

bpb

Zero Trust Security

A Complete Guide

NIKE ANDRAVOUS

bpb

Zero Trust Security

A complete guide

Nike Andravous

www.bpbonline.com

FIRST EDITION 2022

Copyright © BPB Publications, India

ISBN: 978-93-5551-251-2

LIMITS OF LIABILITY AND DISCLAIMER OF WARRANTY

The information contained in this book is true to correct and the best of author's and publisher's knowledge. The author has made every effort to ensure the accuracy of these publications, but publisher cannot be held responsible for any loss or damage arising from any information in this book.

All trademarks referred to in the book are acknowledged as properties of their respective owners but BPB Publications cannot guarantee the accuracy of this information.

To View Complete
BPB Publications Catalogue
Scan the QR Code:

www.bpbonline.com

Dedicated to

My family

About the Author

Nike Andravous is Compliance, Cloud Security, and Zero Trust Thought leader, author, and security and technology strategist focusing on assisting businesses of all sizes in securely adopting new business models. I've spent the last ten years working with clients, vendors, service integrators, and thought leaders to assist educate and accelerating safe cloud adoption, with an emphasis on Zero Trust frameworks and compliance.

About the Reviewer

Robert Forbes is an Enterprise architecture, solution engineering, and software architect. He has successfully helped multiple customers in the design and execution of their corporate IAM strategies, in ways that correspond with both security and business objectives, having over 25 years of industry expertise.

Acknowledgments

There are a few people I want to thank for the support they have given me during the writing of this book. First and foremost, I would like to thank my parents for continuously encouraging me to write the book. I could have never completed this book without their support.

My gratitude also goes to the team at BPB Publications for being supportive enough to provide me quite a long time to finish the book and also giving us the opportunity and providing us the necessary support in writing this book.

We would like to thank our family members for the support they have provided for us to focus on the book during our personal time.

Preface

Chapter 1: Introduction to Enterprise Security, you will get a strong grasp of what Zero Trust is and the information you'll need to effectively navigate your company's particular route to Zero Trust

Chapter 2: Get to know Zero Trust, you will get to know the history and few basic information about Zero trust

Chapter 3: Architectures with Zero Trust, understand about the architecture, and learn about NIST and policy models.

Chapter 4: Zero Trust in Practice, know some real-world instances of Zero Trust systems now that we've covered the basics of Zero Trust.

Chapter 5: Identity and Access Management (IAM), know about the main components of IAM, which are the foundation for all of the book's discussions regarding Zero Trust.

Chapter 6: Network Infrastructure, get an understanding about the firewalls, and gateways.

Chapter 7: Network Access Control, Understand Network Access Control, its value and importance and know how it got replaced by zero trust by it ability

Chapter 8: Intrusion Detection and Prevention Systems, learn about IPS, IDS and IDPS, along with their varieties and comparing them.

Chapter 9: Virtual Private Networks, Learn about the history and types of VPNs. Understand how zero trust and VPN work together

Chapter 10: Next-Generation Firewalls, this chapter will primarily focus on the role of Next-Generation Firewalls (NGFWs) in a Zero Trust environment, understand where and how NGFW solutions should fit into your Zero Trust architecture

Chapter 11: Security Operations, learn… Learn about various security operations in zero trust.

Chapter 12: Privileged Access Management, understand about the IT security business that uses a collection of security functions and procedures to govern, manage, and report on how privileged users (system administrators) access systems and resources

Chapter 13: Data Protection, we'll talk about the data lifecycle, data protection, and data consumption (including labeling and categorization) in this chapter, and how data security fits into a Zero Trust approach.

Chapter 14: Infrastructure and Platform as a Service, the principles and concepts covered in this chapter should help you understand how to secure IaaS and PaaS installations as part of your Zero Trust project, as well as how to enable service mesh-based applications.

Chapter 15: Software as a Service (SaaS), learn how SaaS and Zero Trust security could be implemented.

Chapter 16: IoT Devices, Get to know about installation and set up.

Chapter 17: A Policy of Zero Trust, gives you a framework and structure for thinking about the breadth of your Zero Trust system and evaluating vendor platforms. Because the breadth and depth of Zero Trust systems may be practically endless, it's critical that you have a clear understanding of what should and shouldn't be included, as well as what's appropriate to include in a policy model.

Chapter 18: Zero Trust Scenarios, we'll look at seven potential scenarios in this chapter and talk about how to evaluate and approach them for inclusion in your Zero Trust programme.

Chapter 19: Creating a Successful Zero Trust Environment, this lesson will answer a lot of question and prove that Zero trust environment is simple and helpful

Coloured Images

Please follow the link to download the
Coloured Images of the book:

https://rebrand.ly/0a6dbf

We have code bundles from our rich catalogue of books and videos available at **https://github.com/bpbpublications**. Check them out!

Errata

We take immense pride in our work at BPB Publications and follow best practices to ensure the accuracy of our content to provide with an indulging reading experience to our subscribers. Our readers are our mirrors, and we use their inputs to reflect and improve upon human errors, if any, that may have occurred during the publishing processes involved. To let us maintain the quality and help us reach out to any readers who might be having difficulties due to any unforeseen errors, please write to us at :
errata@bpbonline.com
Your support, suggestions and feedbacks are highly appreciated by the BPB Publications' Family.

Piracy

If you come across any illegal copies of our works in any form on the internet, we would be grateful if you would provide us with the location address or website name. Please contact us at **business@bpbonline.com** with a link to the material.

If you are interested in becoming an author

If there is a topic that you have expertise in, and you are interested in either writing or contributing to a book, please visit **www.bpbonline.com.** We have worked with thousands of developers and tech professionals, just like you, to help them share their insights with the global tech community. You can make a general application, apply for a specific hot topic that we are recruiting an author for, or submit your own idea.

Reviews

Please leave a review. Once you have read and used this book, why not leave a review on the site that you purchased it from? Potential readers can then see and use your unbiased opinion to make purchase decisions. We at BPB can understand what you think about our products, and our authors can see your feedback on their book. Thank you!

For more information about BPB, please visit **www.bpbonline.com.**

Table of Contents

CHAPTER 1

Introduction to Enterprise Security

Enterprise security is difficult to achieve. This is owing to the inherently adversarial character of information security, as well as the complexity of IT and application infrastructures, the breadth and velocity of user access, and the inherently adversarial nature of information security. It's also because most business networks are simply too open, and organizations are leaving themselves tremendously susceptible to assaults by not enforcing the concept of least privilege at both the network and application levels. This is true for both internal networks and public Internet-facing remote access services like Virtual Private Networks, which are vulnerable to any Internet opponent. You'd never choose to create a system like this in today's dangerous scenario. Traditional security and networking solutions, however, which are still widely used, maintain this mindset.

The subject of this book, Zero Trust security, alters that by introducing a new approach to security that enforces the idea of least privilege for networks and applications. Unauthorized users and systems will have no access to any company resources, while authorized users will have just the bare minimum. As a consequence, businesses have become safer, more secure, and more resilient. Through the automatic enforcement of dynamic and identity-centric access controls, Zero Trust improves efficiency and effectiveness.

Please note that the "*zero*" in Zero Trust is a bit of a misnomer—it refers to zero inherent or implicit trust, not literally "zero" trust. Zero Trust is all about methodically laying a foundation of trust and then expanding on it to provide the correct degree of access at the right moment. It might have been named *earned trust*, *adapted trust*, or *zero implicit trust*, and these names would have been more appropriate, but "Zero Trust" has more oomph, and it has stuck. Please don't take "zero" literally!

Zero Trust is a significant and well-known trend in the information security business, and while it has become a marketing phrase, we feel it has actual substance and value.

The goal of Zero Trust is to radically alter the underlying concept and approach to enterprise security, moving away from old and clearly unsuccessful perimeter-centric techniques and toward a dynamic, identity-centric, and policy-based approach.

It can be difficult to make such a change. Enterprise security assets such as directories, IAM systems, IDS/IPS, and SIEM have already been implemented and operationalized in your business and altering things might be challenging. Zero Trust Security is the only company that covers the whole range of corporate security and IT infrastructures, giving detailed architectural counsel and technical analysis with the purpose of speeding up your company's road to Zero Trust. Zero trust in your surroundings, and to act as a guide for you on your path. We think that approaching and achieving enterprise security through Zero Trust is a better and more effective strategy. In some respects, Zero Trust has been linked with network security, and while networks are an important part of Zero Trust, we'll be looking at the complete scope of Zero Trust security, which includes apps, data, identities, operations, and policies.

You have a responsibility as a security leader to push, pull, and prod your business to embrace this new strategy, which will increase your organization's resiliency while also allowing you to advance professionally.

This book, which serves as your guide, is organized into three sections. Part I introduces Zero Trust concepts and defines the framework and terminology that will be used to define Zero Trust and align IT and security infrastructure in Part II. These are the building blocks for telling the whole Zero Trust tale, in our opinion.

Part 2 delves deeper into IT and security technologies, as well as their connections to Zero Trust. This is where you'll learn how to implement Zero Trust in your business, as well as how to adapt and integrate your present IT and security infrastructure into a more contemporary design. We'll look at how different technologies may start to include and profit from identity context to become more successful since Zero Trust takes an identity-centric approach to security.

Part III ties everything together, expanding on the conceptual basis and in-depth technological discussion laid forth in the previous two sections of the book. This section addresses what a Zero Trust policy model should look like, as well as particular Zero Trust situations (use cases) and a strategic and tactical approach to making Zero Trust a success.

It's also worth noting that we're not assessing vendors or vendor products as part of the scope of this book. Our sector changes too fast—the rate of the invention is high—and any such assessments would be obsolete in a matter of months. Rather, we'll concentrate on architectural principles from which you may derive requirements and assess vendors, platforms, solution providers, and methods.

By the end of this book, you should understand that there is no one-size-fits-all solution to Zero Trust. When planning a Zero Trust effort, security directors must consider existing infrastructures, priorities, employee skills, budgets, and schedules. While this may make Zero Trust appear complex, its breadth of coverage really aids in the simplification of business security and architecture. It normalizes things and allows you a centralized approach to design and enforce access policies across a dispersed and heterogeneous infrastructure as an overlay security and access paradigm.

Finally, the purpose of this book is to provide you with a strong grasp of what Zero Trust is and the information you'll need to effectively navigate your company's particular route to Zero Trust.

We've succeeded in our efforts if you come away with this. Let's embark on our journey.

CHAPTER 2

Get to Know Zero Trust

We'll present Zero Trust as a concept, a philosophy, and a structure in this chapter. We'll give a quick review of Zero Trust's history and progress, as well as introduce some guiding principles. We feel that there are basic and extended concepts that are similar to all Zero Trust initiatives and that you should be aware of as you begin your journey.

Structure

- Evolution and history
- The **Zero Trust eXtended** (**ZTX**) model from Forrester research

Objective

This chapter's purpose is to provide you with a practical definition of Zero Trust based on these concepts, as well as a set of core platform needs.

Evolution and history

In a conventional *castle wall and moat* strategy, security barriers were traditionally set at the border of the company network. Remote employees and remote tasks grew increasingly popular as technology advanced. The security perimeter had to stretch beyond the corporate boundary to include the devices and networks from which the remote user was connected, as well as the resources to which they were connecting. With varying degrees of success, security and network teams were obliged to meet these business objectives and alter the methods by which enterprises implemented security and access.

In the important *No More Chewy Centers: Introducing the Zero Trust Model of Information Security* whitepaper published in 2010, Forrester Analyst *John Kindervag* coined the phrase “Zero Trust.” This report encapsulated themes that had been explored in the business for a few years, with the Jericho Forum playing a key role. The trend away from a hard perimeter, according to the Forrester report, was toward a strategy that required evaluating and comprehending network pieces before they could gain a level of confidence and access. Forrester eventually refined this notion into the ZTX framework, which covers data, workloads, and identity as fundamental components of Zero Trust. Around the same time, Google started their internal **BeyondCorp** program, which built a version of Zero Trust and put core Zero Trust pieces in place, thereby removing their business network boundaries. Beginning in 2014, Google affected the industry by publishing a series of papers outlining their ground-breaking internal implementation. The Cloud Security Alliance also launched the **Software-Defined Perimeter** (**SDP**) architecture in 2014, which gave a real design for a security system that adhered to Zero Trust principles.

Later in *Chapter 4: Zero Trust in Practice*, we’ll look at both BeyondCorp and SDP through the prism of Zero Trust.

Gartner, a market research business, updated and altered its **Continuous Adaptive Risk and Trust Assessment (CARTA)** approach in 2017, which shares many aspects with Zero Trust. CARTA incorporates risk and posture-related with identity and devices accessing the environment, in addition to identity and data aspects.

In 2020, the US **National Institute of Standards and Technology** (**NIST**) produced a Zero Trust Architecture paper and a related US National Cybersecurity Center of Excellence initiative, reinforcing the industry’s focus on Zero Trust.

Zero Trust is still evolving as manufacturers and standards bodies examine and modify specifications and implementations, seeing it as a major shift in the way people think about information security. Finally, the industry has agreed that these modifications and enhancements are required to prevent hostile actors from gaining access to private resources within organizations, exfiltrating data, and disrupting operations.

We, the book's writers, are both information security specialists who spend a lot of time talking to security professionals about Zero Trust. *What's new about Zero Trust—how is it different from what's already been done?* It is a regular question we get. Although certain aspects of Zero Trust, such as least privileged access and role-based access control, are ideas that are widely used in today's networking and security infrastructure (and must be used in Zero Trust settings), they are insufficient to complete the picture.

Prior to Zero Trust, foundational security features frequently only accomplished coarse-grained isolation of people, networks, and applications. In most firms, development and production environments are separated, for example.

Zero Trust, on the other hand, compounds this by virtually demanding the separation of all identities and resources. Zero Trust is an automated platform that offers fine-grained, identity-and-context-sensitive access restrictions. Despite the fact that Zero Trust began as a narrowly focused method of not trusting any network identities unless they were validated and permitted, it has evolved in breadth to provide a far broader range of security capabilities throughout an organization's environment.

Before we discuss what, we feel are the core Zero Trust concepts, let's have a look at the Forrester and Gartner Zero Trust models.

The Zero Trust eXtended (ZTX) model from Forrester research

Forrester first launched their Zero Trust model in 2010, and it has since been updated and re-issued as **Zero Trust eXtended** (**ZTX**). As seen in *figure 2.1*, ZTX offers more material and a well-rounded model that prioritizes data.

This underlines Forrester's opinion that data explosion is at the heart of what has to be secured in both on-prem and cloud systems. Workloads, networks, devices, and people are all conduits for data, and as such, they must be protected as well. Let's take a look at each of these components one-by-one:

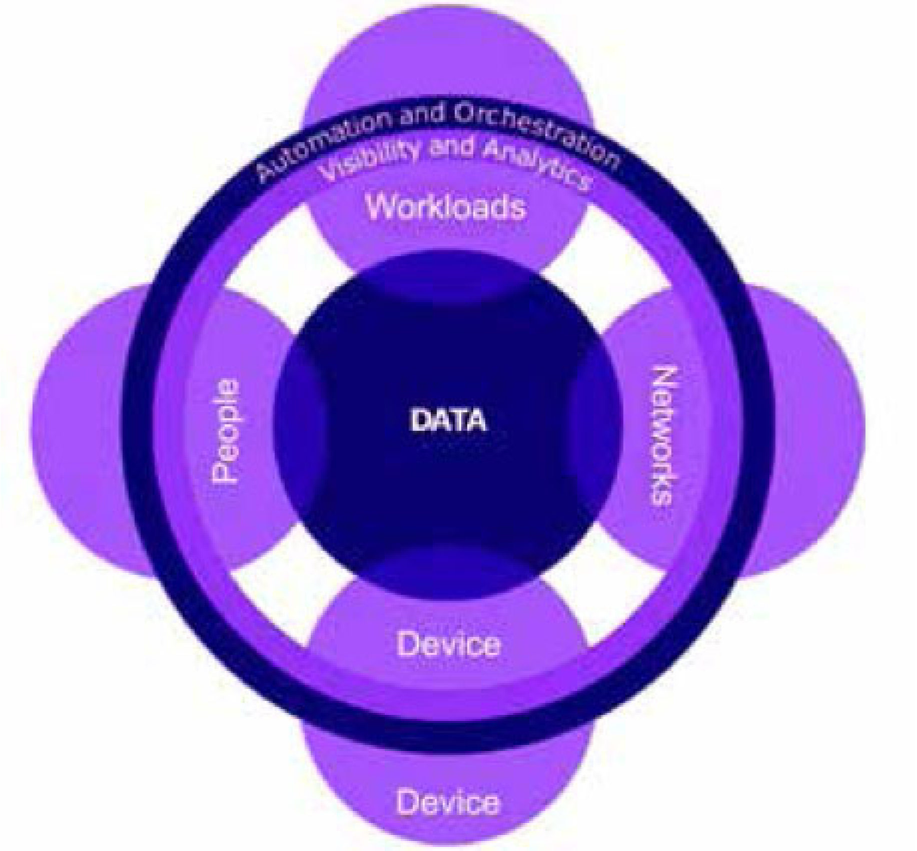


Figure 2.1: *Forrester Zero Trust eXtended Model (Credits: https://www.guardicore.com/zero-trust-security/)*

Data is at the heart of the ZTX model, and it contains data classification and protection at the core of the needs to enable the Zero Trust Model, which Forrester also labels as "*value*" to emphasize its importance. We treat data as one of the Resources that Zero Trust systems must safeguard throughout the book. **Data Loss Prevention** (**DLP**) should also be a component of a Zero Trust architecture and should be linked to the policy model with the ability to apply contextual access controls whenever possible.

Networks: The ZTX model's Network pillar is largely concerned with network segmentation, both from a user and server standpoint, in order to provide stronger security based on identity-centric features. It's vital to remember that traditional network security architecture includes many current components, such as **Next-Generation Firewalls** (**NGFWs**), **Web Application Firewalls** (**WAF**), **Network Access Control** (**NAC**) solutions, and **Intrusion Protection Systems** (**IPS**).

In a Zero Trust system, all of these elements have a role to play. In Chapter 3*: Architectures With Zero Trust*, we'll introduce these components in the context of typical business architecture, and in Part II of the book, we'll go through their connection with Zero Trust in detail.

Multiple parts of **Identity and Access Management** (**IAM**) must be included in the people pillar of the ZTX model. **Role-based access control** (**RBAC**) and **attribute-based access control** (**ABAC**) are well-known IAM approaches, and Zero Trust makes them more widely and effectively applicable throughout the corporate infrastructure. Another need is **Multi-Factor Authentication** (**MFA**), which is required to support Zero Trust. Finally, another key component of the people pillar is **Single Sign On** (**SSO**), which uses current, open standards like OAuth and SAML. As you'll see throughout this book, we believe that Identity should be at the heart of any Zero Trust ecosystem.

Containers, apps, infrastructure, processes, and other components that make up the logical operations that drive business throughout both client-facing and backend business systems are referred to as workloads by Forrester. Workload access constraints based on metadata must be applied consistently across hybrid environments if Zero Trust is to be achieved. In *Chapter 17: A Policy of Zero Trust*, we'll go more into this topic.

Devices: The identification, inventory, isolation, security, and control of the device should all be included in the security model.

In *Chapter 3: Architectures With Zero Trust*, we'll look at device-based user agents and how they fit into the Zero Trust ecosystem. Later in *Chapter 4: Zero Trust in Practice*, we'll look at how gadgets played a role in Google's BeyondCorp implementation.

Within ZTX, visibility and analytics refer to the consumption and display of data across the organization in order to support educated security choices based on contextual data. We believe that this is crucial, especially when it comes to combining data from diverse sources. There is currently no one platform that encompasses the required range of capabilities, but this is an ever-evolving sector. We'll go over this in more detail in *Chapter 11: Security Operations*.

Automation and Orchestration: Within ZTX, automation and orchestration are needed to automate manual activities and link them to security policies and reaction actions. This feature, we feel, is vital to the success of a Zero Trust platform—Zero Trust is fundamentally dynamic and flexible, and the only way to achieve this is through enterprise-wide automation and orchestration. We'll go over this in more detail later because automation is one of our main Zero Trust concepts.

Gartner's Zero-Trust strategy

Gartner has used a **Continuous Adaptive Risk and Trust Assessment** (**CARTA**) approach to Zero Trust. CARTA's concept is to provide continuous risk assessment for individuals, devices, applications, data, and workloads from a predict, prevent, detect, and respond viewpoint.

CARTA employs the core process of implementing a security posture, monitoring it, and adjusting it across several security planes. These principles, according to Gartner, should be applied throughout the whole company, including security, policy, and compliance needs. Gartner uses the words **Zero Trust Network Access** (**ZTNA**) for user-to-server security and **Zero Trust Network Segmentation** (**ZTNS**) for micro-segmentation/server-to-server security to describe Zero Trust. Their general security structure is based on CARTA, and its concepts are quite similar to what we advocate for here. It doesn't matter if your strategy endeavor is called Zero Trust, CARTA, Earned Trust, or something else at the end of the day.

Gartner's CARTA concepts and aims are valid, and we feel they are in line with the ones we're discussing in this book.

Our point of view on zero trust

Zero Trust is a comprehensive approach to safeguarding network, application, and data resources, focusing on an identity-centric policy architecture for access management. All businesses have a set of IT and security systems in place, but Zero Trust requires that they be seen and managed holistically, with identity at the center and the ability to implement attribute - and context-sensitive policies across the board. As we look at the basic concepts of Zero Trust, which we've divided into core and expanded principles, this should become evident.

The fundamentals

There are three key Zero Trust concepts that are widely acknowledged as foundational and necessary in the business. These were first outlined in Forrester's *No More Chewy Centers* paper, and we feel they must be followed in any Zero Trust deployment. We've included the concepts mentioned in the NIST Zero Trust architecture document in addition to these key principles. Here's our take on it, as seen through the eyes of the existing industry.

Ensure that all resources, regardless of location, are accessible securely. This is a succinct, strong phrase that embraces numerous aspects. For starters, it necessitates including all resources in the scope of a Zero Trust solution. Implicitly, this necessitates a comprehensive approach to Zero Trust, as well as the elimination of silos and obstacles that have previously existed between security products and teams.

Second, this concept needs Zero Trust to provide secure access to all resources (data, apps, and servers) for all identities (human and machine), independent of the identity's location or the location or technology of the resource being accessed.

This concept basically dictates the demise of the existing corporate perimeter and the replacement of it with a new security paradigm. It also means that not only must network traffic be encrypted when it passes across untrusted network areas but that all access must be governed by a policy model that is enforced—the second principle.

Adopt a strategy of least privilege and tightly enforce access control

Prior to Zero Trust, the notion of least privileged access to resources was not new, but it was difficult to implement widely. Using security and identity context, least privilege must be handled consistently across locations and resource kinds, as well as at the network and application layers.

Until now, security solutions haven’t been able to bridge the gap between network and application security. Users (and their devices) formerly had broad network access, and apps depended on authentication-only access control.

The Finance server’s login page was accessible to everybody in the firm, but only Finance users had accounts and passwords. This is no longer an adequate degree of protection.

There are simply too many known and severe vulnerabilities that may be remotely exploited and do not require authentication. We’ll say it again: the ability to transmit network packets to a system is a privilege that must be controlled accordingly. Users must not be able to connect to a service at the network layer if they are not permitted to access it (for example, having credentials to SSH into a server or log in to a VPN).

Examine and record every traffic

Because networks are the way through which scattered components link and communicate with one another, they hold a special place in the security and IT architecture. As a result, the fourth basic concept necessitates network traffic monitoring and logging. As we’ll see in *Chapter 3*: *Architectures With Zero Trust*, zero trust systems are ideally suited to this. They’re often made up of a dispersed collection of network enforcement points.

It’s vital to emphasize that, due to processing and storage costs, Zero Trust systems should review and log network traffic metadata extensively but should be more selective in their inspection of network traffic content. (We’ll go over this in more detail in *Chapter 8*: *Intrusion Detection and Prevention Systems*.)

The Zero Trust system should enrich network traffic information by adding identity and device context, which should then be fed into NGFWs , network monitoring tools, and SIEMs to improve their ability to detect, alert, and respond, as well as support incident response and other alerting mechanisms.

Principles expanded

We think there are three more Zero Trust principles that are vital and necessary in any enterprise-class Zero Trust environment, in addition to the fundamental Zero Trust principles presented. Ascertain that all components provide event and data exchange APIs.

The first key premise of Zero Trust is that it must provide a comprehensive security policy and enforcement architecture that covers all aspects of the IT ecosystem. As a result, it must be able to work with many (preferably all) of the ecosystem's components. It is critical to integrate previously segregated security solutions, infrastructure, and business processes. As we've seen throughout our conversations, combining identity and security solutions allows Zero Trust to create a more secure environment by creating a comprehensive security context.

These integrations will be utilized to trigger and respond to events, as well as exchange data and log information to enable our next concept. This approach has a corollary: every security and IT component you incorporate into your Zero Trust platform increases its value, effectiveness, and reach. Every siloed (non-integrated) component, on the other hand, adds friction, reduces the efficacy of your Zero Trust solution, and can compromise security.

Context and events can be used to automate operations across contexts and systems.

Automation is a critical component of a successful Zero Trust environment, and it is required for even small-scale operations. Zero Trust is built on a collection of dynamic access control rules that adapt to the identity, device, network, and system context.

Zero Trust architectures all need a centralized **Policy Decision Point** (**PDP**) linked to a distributed collection of **Policy Enforcement Points** (**PEPs**) through a logical control channel, as we'll see in *Chapter 3: Architectures With Zero Trust*. This channel is necessary for a Zero Trust system to function since it is used to automate changes to enforced policies via integration/APIs.

In a Zero Trust system, automated modifications to access can take various forms, such as providing access via an identity management system, an access management system, or a network access control system. Other automatic tasks might include the temporary or permanent termination of access to a resource, as a result of a lifecycle management event or a context change, for example.

While automated actions are essential in an operational environment, this does not exclude the need for human intervention or the inclusion of explicit manual steps in a process before beginning an automated reaction. To put it another way, automation does not imply *automation*. To fulfill security and compliance rules, many accesses request processes, for example, require management approval. This procedure necessitates a person reading data, making a choice, and submitting that decision to the system. The only manual step in this procedure should be the provisioning of any access modifications; the remainder of the workflow should be automated.

Bring tactical and strategic value to the table

Ultimately, essential Zero Trust activities must be linked to commercial value. Infrastructures, teams, operations, and the end-user experience can (and often do) suffer substantial consequences as a result of Zero Trust initiatives. Even when the effects are beneficial, technological, cultural, and political changes are typically difficult to achieve. Many components in your environment will be altered or incorporated into your Zero Trust environment as an enforcement point or policy driver, and the changes connected with a Zero Trust project can be far-reaching.

Zero Trust is a journey and a financial and time investment. Understanding your organization’s business drivers and priorities can assist you in justifying and implementing your Zero Trust strategic vision in your enterprise environment. As you begin your trip, you must achieve modest deployments and tactical victories. This will simplify your Zero Trust journey while also increasing internal momentum and support. That is, you will enable your business to achieve its full strategic value by achieving early tactical successes inside the context of your strategic Zero Trust architecture. Each new project that is a success helps to open doors and get support for your Zero Trust effort.

A provisional definition

It’s critical to grasp what Zero Trust is as we progress through this book and present notions of Zero Trust principles, architectures, and functioning examples. We find it helpful to think of Zero Trust as a lens through which security projects and components may be seen and interpreted. To this purpose, we provide the following succinct definition:

A Zero Trust system is an integrated security platform that leverages contextual data from identity, security, and IT infrastructure, as well as risk and analytics tools, to inform and allow the consistent application of security rules across the company. Zero Trust transforms security from a perimeter-centric approach that is ineffectual to a resource and identity-centric approach.

As a consequence, businesses may adjust access restrictions to a changing environment in real-time, resulting in greater security, lower risk, simpler and more robust operations, and increased business agility.

In addition to the concepts, we stated before, this core definition enables us to propose the first set of Zero Trust criteria, which we will address next. As a result, firms may alter access limits in real-time in response to changing conditions, resulting in better security, reduced risk, simpler and more robust processes, and increased business agility.

This basic definition, in addition to the previously mentioned notions, allows us to present the first set of Zero Trust requirements, which we will discuss next.

Platform requirements for Zero Trust

We propose a baseline set of platform requirements in this part, which are derived from the Zero Trust principles previously addressed. This section’s objective is not to just reiterate the principles, but to try to emphasize key features from a platform standpoint. Some of these concepts (particularly APIs and Integration) are better described as criteria for specific IT and security roles, but we’ve defined them broadly in general:

- Encryption is required for data plane connections. Any exclusions must be planned ahead of time (for example, DNS).

- Access restrictions for all sorts of resources must be enforced by the system. Identity centric and contextual policies must drive access control systems.
- Identity and contextual policies should be able to regulate access to data resource safeguards.
- The system and policy model must be capable of securing all users at all times and in all places. For remote and on-premises users, the policy model and controls must be consistent.
- Devices must be able to be assessed for security posture and configuration before being permitted access, as well as on a regular basis afterward.
- BYOD devices must be distinguished from corporate-managed devices, and access levels must be controlled accordingly.
- The policy must clearly permit access to any network resource.
- No user or device should have unrestricted network access by default.
- Access controls must be able to discriminate between distinct network resources and services. Access to HTTPS, for example, must be allowed separately from access to SSH.
- The business policy must be followed when granting access to specific data pieces located within applications or containers with various classifications.
- Metadata about network traffic must be logged and supplemented with identification context.
- The ability to inspect network traffic for security and data loss is required.
- Access control policies specified by on-premises systems should be applied to workloads transported to the cloud.
- Identity-centric details must be included in automation.
- To offer an efficient and effective incident response, automation must contain identity-centric details.
- For effective and dynamic policy enforcement, logs must be integrated in analytics tools.

Conclusion

In this chapter, we looked at the history of Zero Trust, starting with Forrester's introduction of the phrase in 2010, and how it has evolved over time thanks to companies like Google, NIST, CSA, and others. We discussed and modified three key Zero Trust principles, as well as introduced three expanded concepts, based on this historical context. We feel that this collection, taken collectively, should be the foundation of any Zero Trust campaign.

We'll introduce a typical enterprise architecture model in the upcoming chapter. It won't be exhaustive, but it will serve as a starting point for discussing Zero Trust deployment methods and how they integrate within the company.

Later in the book, in Part II, we'll look at how Zero Trust affects IT and security systems.

CHAPTER 3

Architectures With Zero Trust

So far, we've covered the history of Zero Trust, our perspective on it, and the basic set of principles that define it. Zero Trust is a principle that may be applied to a wide range of architectures (as well as a wide range of commercial goods). It will become evident that there is no one-size-fits-all architecture, and that each business must assess its own unique requirements in order to strategically build the best method for its Zero Trust journey.

It's impossible to design a "one-size-fits-all" Zero Trust architecture due to the vast range of methods and the uniqueness of each organization's starting point.

Regardless, we've accepted the challenge and are attacking it in two ways. First, we'll introduce and explore a simplified but typical corporate architecture that we'll introduce and explore in this chapter. This architecture is meant to be representative of a typical business, not a precise or thorough technical representation of any specific company or network. Its purpose is to demonstrate a basic visual model of an architecture that has many aspects in common with most companies, as well as links and interconnections between these diverse components.

We'll go through each of the IT and security components used once we present the corporate architecture. This allows us to prepare them for our in-depth look at them in Part II of the book. We'll look at how each one relates to, integrates with, and should be seen through the lens of a Zero Trust architecture.

Second, in this chapter, we'll provide a Zero Trust architecture conceptual model. This is also an issue since different approaches to Zero Trust depend on the underlying enterprise architecture and the decisions taken by enterprise security architects.

We’ll start with the US National Institute of Standards and Technologies (NIST) Zero Trust Design from Special Publication 800-207 for our Zero Trust architecture.

However, we’re expanding and improving that architecture to make it more useful to businesses and to better connect it with our strategy. That is, we will use these architectural principles throughout the book to make Zero Trust notions real and applicable to your business.

Consider this: your company network and security architecture includes a variety of components such as firewalls, network access control, intrusion detection and prevention systems, and so on. In a Zero Trust architecture, the most of these will continue to exist (although some may not). However, with Zero Trust, the way your infrastructure parts are designed and operated should change, resulting in greater security and simpler operations in all circumstances. Let’s begin with an overview of corporate architecture.

Structure

You will learn following chapters:

- An example of enterprise architecture
- Identity and Access Management (IAM)
- Privileged Access Management (PAM)
- Network Access Control (NAC)
- Intrusion detection/prevention
- Virtual Private System (VPS)
- Security information and event management (SIEM)
- Web application and web server firewall
- Infrastructure as a Service (IaaS)
- Cloud Access Security Brokers (CASB) and Software as a Service (SaaS)
- A Zero Trust architecture
- The NIST Zero Trust model
- A Zero-Trust architecture in concept

- Models of Zero Trust deployment
- Model for Cloud-Routed Deployment
- Deployment model for microsegmentation

Objective

You will get to learn about the US National Institute of Standards and Technologies,

An example of enterprise architecture

The *figure 3.1* depicts logical connections between essential networking components in a business design encompassing the most typical IT and security infrastructure aspects. We've left out a lot of information from this diagram for clarity's sake; we'll go through each of the components, as well as their interdependencies and linkages, in the corresponding chapters of Part II of the book. For now, we'll give a quick overview of each part, emphasizing its purpose in the architecture, how our fictitious company uses it, and how they want to enhance it. Let's take a quick look at the visual components we'll be employing throughout the book:

A dashed line represents logical links between things. A solid bolded line denotes a secure (encrypted) link between items. The data flow between objects in the diagram utilizing native application protocols is represented by a solid line (not bolded) (which may or may not be encrypted). The "*R*" represents the resources being accessed (workloads, services, or data). Finally, ellipses between resources indicate a collection's shared set of resources.

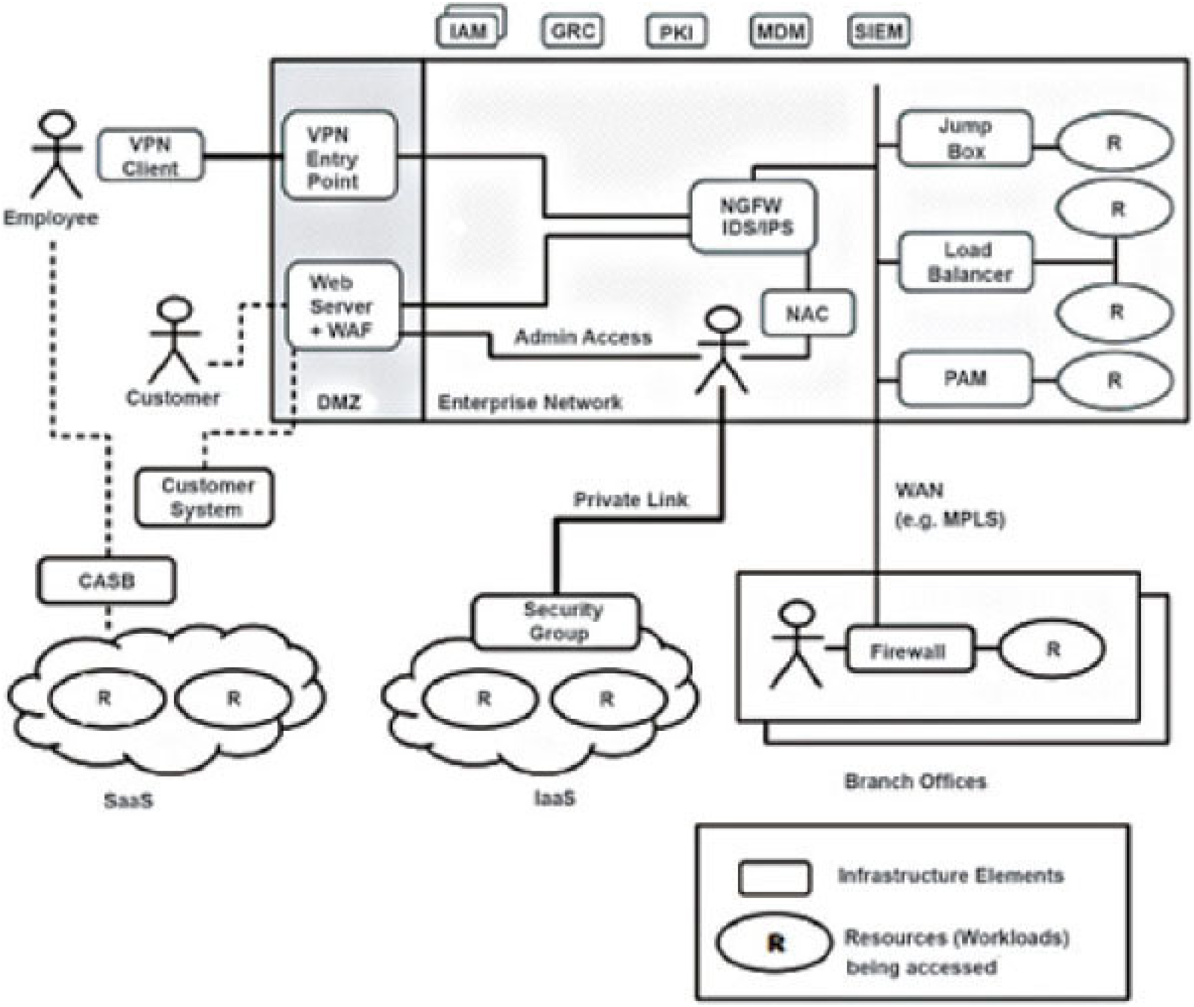


Figure 3.1: *A representative enterprise architecture*

In our architecture, the company has a main headquarters network as well as several branch offices. Users must be able to securely access a number of networked resources (workloads) housed in each of these physical locations, illustrated as R.

On addition, this company has workloads operating in private networks on a public **Infrastructure as a Service** (**IaaS**) provider, as well as many **Software as a Service** (**SaaS**) resources accessed by various user groups. This company, like most others, has a diverse set of access control and networking techniques, as well as an ecosystem of IT and security infrastructure components.

In the parts that follow, we'll go through why and how they use each of them, as well as how they want to improve.

Identity and Access Management (IAM)

This company has numerous identity suppliers, which is a frequent occurrence in today's business world. In this example, the company has one main **Identity and Access Management** (**IAM**) system, but there are multiple smaller ones that are still in operation, owing to several corporate acquisitions. Their identification and authentication systems are used to manage users—mostly workers and some contractors. They have a mix of Multi-Factor Authentication (MFA) and a rudimentary identity governance mechanism in place. Their most recent audit revealed a number of medium-priority issues that they must solve.

They do have a strategy to rationalize and centralize various IAM systems, but there are a number of stumbling blocks, including IAM connections with apps, as well as automated and manual provisioning procedures. Whatever security enhancements or modifications they make will have to work with their existing (messy) IAM infrastructures—waiting for it to be rationalized or centralized first is unrealistic.

Even in their present state, they have a decent set of roles in place thanks to **Role-Based Access Control** (**RBAC**) tools and processes, and they naturally want to extract more value out of them. Their present security architecture, on the other hand, is mainly unconnected to their IAM systems. They understand that this is a source of friction, expense, inefficiency, and ineffectiveness, and they want to address it as they progress toward Zero Trust.

They do have a strategy to rationalize and centralize various IAM systems, but there are a number of stumbling blocks, including IAM connections with apps, as well as automated and manual provisioning procedures. Whatever security enhancements or modifications they make will have to work with their existing (messy) IAM infrastructures—waiting for it to be rationalized or centralized first is unrealistic.

Even in their present state, they have a decent set of roles in place thanks to RBAC tools and processes, and they naturally want to extract more value out of them. Their present security architecture, on the other hand, is mainly unconnected to their IAM systems.

More dynamic (ephemeral) workloads operating in on-premises containerized or virtualized systems, along with increased remote user access, has made these solutions less successful overall. They’re essentially compelled to allow overly wide network access since existing security tools —which were created and constructed to safeguard a more static and predictable IT infrastructure—are increasingly unable to discern between various users and target workloads.

This open network access is now a top concern for them, as they recently had a malware assault that extended widely throughout the network and affected a large number of computers. They’re also turning to Zero Trust to help them better unify access restrictions throughout their network.

More dynamic (ephemeral) workloads operating in on-premises containerized or virtualized systems, along with increased remote user access, has made these solutions less successful overall. They’re essentially compelled to allow overly wide network access since existing security tools —which were created and constructed to safeguard a more static and predictable IT infrastructure are increasingly unable to discern between various users and target workloads.

This open network access is now a top concern for them, as they recently had a malware assault that extended widely throughout the network and affected a large number of computers. They’re also turning to Zero Trust to help them better unify access restrictions throughout their network.

Jump boxes

This business has been employing jump boxes (also known as jump hosts or leap servers) as hardened access points for limiting admin access to high-value resources. Production systems and backup systems, for example, are network assets. On a different network section, it’s separated. Despite the fact that a previous audit revealed and compelled. They needed to solve certain security concerns with their jump boxes (shared credentials, for example). Despite these obstacles (including a lack of MFA), they still have a lot of difficulties to overcome. The fact that the jump boxes have full network access to the high-value data is one of them. Inability of systems to enforce the intended business process (request and approval). Their lack of integration with identity systems, as well as transitory access.

Privileged Access Management (PAM)

For access to many high-value systems, this company uses a **Privileged Access Management (PAM)** solution to facilitate password vaulting and offer session recording. PAM is being utilized rarely across the company because to its expense and complexity, despite the fact that it provides a safe method to obfuscate passwords and secure access to certain systems inside the enterprise.

The PAM solution currently only provides limited contextual awareness and has no connectivity to their core Identity solution. This restricts its ability to provide a role-based solution for determining who should have access to the PAM solution's high-value systems.

In an ideal world, the PAM solution would be able to make access control choices based on contextual data in accordance with policy and compliance standards. As part of their comprehensive reevaluation of their usage of jump boxes and PAM for access to high-value resources, integrating it with their IAM solution is also a priority.

Network Access Control (NAC)

This company employs a **Network Access Control** (**NAC**) solution to manage network access for on-premises users as well as guest Wi-Fi access at their headquarters office. To identify authorized devices and assign them to VLANs, this hardware-based solution, which is part of the networking infrastructure, employs enterprise-issued certificates.

This worked pretty well for the enterprise headquarters at first, but because NAC is operationally difficult, it hasn't kept up with networking improvements, and users have access to a larger selection of workloads and data than expected. The primary reason of a recent audit finding regarding network access to production systems was this *drift*.

Furthermore, their NAC solution is a silo in more ways than one. First, they decided not to implement NAC in their branch offices, owing to the high cost and complexity. As a result, different sorts of access controls are applied to people in those offices. Second, NAC is manifestly ineffective in remote access and cloud situations. Separate access policy models exist in those situations, which are coarse-grained and static.

Finally, their NAC hardware is nearing the end of its useful life. After considering all of these factors, the enterprise's security officials have opted to deactivate the NAC. This will allow them to reallocate NAC funds to their Zero Trust effort while also updating their infrastructure and lowering complexity and operating costs.

Intrusion detection/prevention

It is a term used to describe the process of detecting and preventing intrusions. This organization, like many others, has a network-based **Intrusion Detection System/Intrusion Prevention System** (**IDS/IPS**) in place, which is implemented using a combination of modules operating within their **Next-Generation Firewalls** (**NGFWs**) and some open-source IDS/IPS implementation. These systems are utilized by their **Security Operations Center** (**SOCs**) to detect unusual behavior and respond to it using a combination of automated and manual actions.

However, as the size and complexity of the network has grown, as has the use of cloud-based services where their IDS can't be an inline *chokepoint*, and the rising usage of encrypted protocols, their IDS has been less effective.

They desire a more thorough and integrated approach to recognizing and responding to signs of compromise in their diverse environment. They believe they are wasting too much time and money while getting just mediocre outcomes. They'd want to have a single location to create IDS policies and many places to enforce them across their network, user devices, and workloads in the ideal world. They also seek to get both quantitative (less noise and false positives) and qualitative (better context for security analyst decision-making) benefits.

Virtual Private System (VPS)

Within this setting, their **Virtual Private Network** (**VPN**) is the sole technology for remote access currently in use. This VPN has been set place to provide distant workers access to the corporate network. However, the firm has suffered performance and reliability challenges as a result of an expanding remote workforce and growing security concerns, and the VPN gives little information regarding identities within the environment. Furthermore, the company is concerned about the VPN solution's primitive level of access controls, which allow remote users to travel freely around the network with little restrictions.

The firm would wish to raise the security context while reducing the possibility for service and performance effects. They want to integrate remote access with their identity provider and leverage user and device attributes for access decision-making and enforcement in order to deliver this security.

Next-Generation Firewalls (NGFWs)

Traditional firewall features, IDS/IPS, certain application awareness and control capabilities, and a remote access VPN are all included in this company's NGFWs (discussed separately earlier). They're not unhappy with their primary NGFW, but they're having trouble with a hybrid infrastructure —they have two enclaves of various vendor NGFWs, and they've never been able to justify the capital and operating costs of consolidating them into one vendor.

As a result, their capabilities differ across these two enclaves, causing operational friction, mismatched policy, and control models, as well as technological challenges when traffic must pass between these security zones.

They want a security and operational solution that gives them a uniform policy model that works consistently across all of these physical infrastructures. They want to avoid the cost of updating this gear because they still have a lot of capacity and depreciation left. They want to start integrating threat information into their security product as well.

Security information and event management (SIEM)

This company employs a hybrid **security information and event management** (**SIEM**) solution that combines classic on-premises SIEM with a newer cloud-based SIEM. They want to shift completely to a cloud-based SIEM, but they have certain on-premises systems that they've modified and integrated. This integration is rigid and difficult to maintain, despite its practical importance.

Migrating to a cloud based SIEM will provide them improved speed and scalability, as well as the opportunity to take use of the more contemporary platform's features, such as the ability to better integrate log data from a wider range of sources.

They want to improve the data that the SIEM can utilize and use it to influence user access through a risk score measurement.

Overall, they're happy with their cloud-based SIEM, but they'd like it to have more *teeth*—that is, the ability to change user access automatically. They plan to do so by combining their Zero Trust programme with the implementation of a **Security Orchestration, Automation, and Response** (**SOAR**) system.

Web application and web server firewall

An online-facing system utilized by their business customers—via a web portal and a set of web APIs—drives a substantial portion of this company's income. This system is located in the enterprise network's DMZ and is linked to a number of production systems. It's safeguarded by a Web Application Firewall (WAF), which protects the web portal and API against application and network-level assaults like SQL injection and cross-site scripting.

This web application has various components that are pertinent here. A public component exists, which is essentially a section of their public website. Everyone, including unauthenticated and anonymous users, should be able to view this. They also offer a free hands-on sample of their service, which is hosted in a sandbox demo tenant. This has shown to be an effective technique for generating new business.

The rest of the website is more private, with only identifiable and authenticated individuals being able to view it. Customers log in and utilize the application to do business with this company using a sophisticated online UI.

This programme also has an API that is frequently utilized by customer systems to transact business—in fact, the API has recently surpassed the UI, generating 75% of their online revenue against 25% for the web UI. They are comfortable with the system's exterior security and do not believe it is necessary to improve it. System administrators have administrative access to the system internally, of course. They have a limited set of access controls in place for these administrators, which they want to strengthen as part of their Zero Trust project.

Infrastructure as a Service (IaaS)

It is a type of cloud. By utilizing **Infrastructure as a Service** (**IaaS**), this organization has improved its compute and networking capabilities and created a "*private link*" tunnel between the on-premises network and the cloud infrastructure; the same flat network approach is maintained even though the infrastructure is now deployed in the cloud. Despite the fact that this enables connection, it does not give any further security. In reality, because they have one network but two different security concepts and tools, this relationship adds to the complexity.

Although the company may use the extra monitoring and networking services provided by the **Cloud Service Provider** (**CSP**), its on-premises infrastructure is still reliant on older networking services, including some layer 2 aspects.

These components will not function in a cloud environment, causing the company to reassess its security strategy for IaaS.

The company wants to deliver dynamic and context-sensitive security that is compatible with and linked with their existing on-premises security approaches. It should be identity-centric, giving them a unified view of access control and monitoring across on-premises and cloud settings. Finally, they want to reproduce the same level of control and automation seen in IaaS across their heterogeneous environment without having to relocate everything to the cloud.

Cloud Access Security Brokers (CASB) and Software as a Service (SaaS)

As the company has evolved, it has made sense to use **Software as a Service** (**SaaS**) solutions to support major business operations such as HR and other tasks. Furthermore, as business units have acquired their own SaaS apps to support expansion, there has been a large increase in resources that are not totally safeguarded (shadow IT). With all of this quick expansion, the company has implemented a **Cloud Application Security Broker** (**CASB**), which has assisted them in locating and securing resources.

The company wants to expand its use of the CASB to not just prevent additional shadow IT, but also to better (and more extensively) implement **Data Loss Prevention** (**DLP**). They also want to be able to manage their SaaS apps in a more secure and identity-centric manner.

This brings us to the end of our overview of this company's present architectural aspects and their plans to develop them. Note that each of these components will be examined in greater depth in Part II of the book, from a Zero Trust perspective. Let's have a look at the Zero Trust architecture's structure and components.

A Zero Trust architecture

We'll provide a notional Zero Trust architecture in this part, which builds on the work described in the NIST article while improving and expanding it. We've taken on the same problem as the NIST writers in this chapter, namely, that Zero Trust begins with a set of principles and a philosophy, and that there are a variety of corporate security architectures that may be utilized to meet Zero Trust's aims.

We recognize that creating or depicting a single architecture that is universally applicable is difficult; instead, our purpose is to offer a collection of architectural components and needs that you may utilize to build a meaningful and beneficial architecture for your business.

The NIST Zero Trust model

NIST proposed a logical collection of Zero Trust components, as depicted in *figure 3.2*, as we described in the previous chapter. This graphic depicts some of the key topics and elements that will be discussed throughout the book:

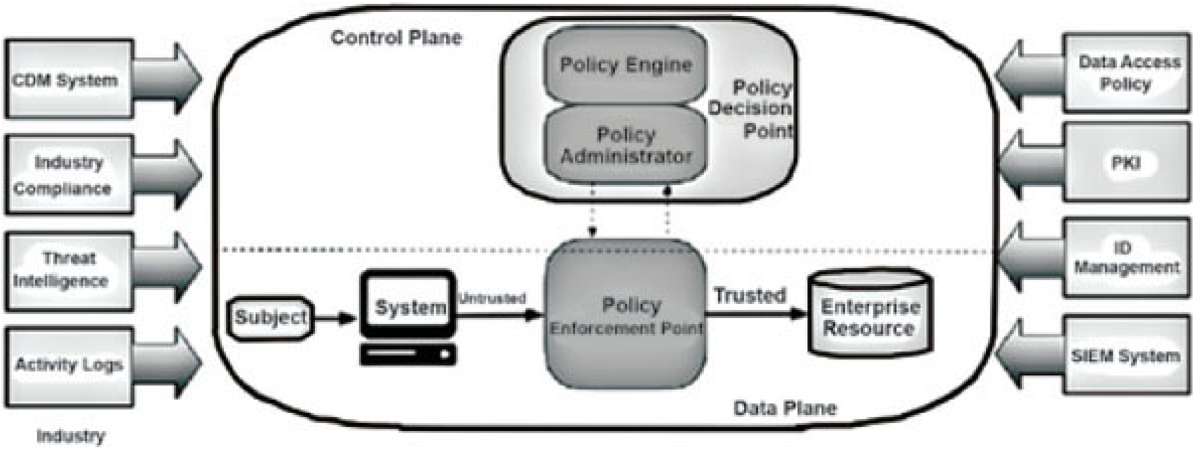


Figure 3.2: *Zero Trust Architecture Core Components. Source: NIST SP 800-207*

The first is the concept of a Subject, which NIST defines as a person, application, or device that operates on (or with) a computer system and has access to an Enterprise Resource. This resource might be an enterprise-controlled programme, data, document, or task that is secured by the Zero Trust system. Throughout the book, we will refer to this as a Resource.

The subject is assumed to be working in an untrustworthy environment on an untrustworthy network, and can only access the Resource through a Policy Enforcement Point (PEP). The implicit trust zone, as defined by NIST, is how the PEP regulates the subject's access to the resource, which we'll go over in more detail later. The Policy Decision Point is in charge of storing and making policy decisions, not the PEP (PDP).

The Subject connects with the Enterprise Resource over the data plane, which is separate from the control plane—the PDP and PEP *communicate on a network that is conceptually independent and not immediately accessible by enterprise assets and resources*, according to NIST. For application data transmission, the data plane is employed."

The extra parts previously portrayed as being outside the system (for example, the CDM and PKI) must be thought of as logically part of any Zero Trust system—or at the very least, with a series of bidirectional arrows denoting varying degrees of integration. These components provide vital context (input) to the Zero Trust system and have a significant impact on its policy decisions. Throughout the book—particularly in the Part II—we'll argue that all of these systems must work together as producers and consumers of data and events. As we investigate how these aspects interact with and impact the PDP and PEPs, there's a lot to speak about. We also want to make sure you understand the main ideas of the PDP and PEPs so you can explore how different aspects of your IT and security infrastructure may and should be considered PEPs inside your yet-to-be-realized Zero Trust architecture. This is why, as shown in *figure 3.3*, we have a variety of conceptual PEPs in place across the company, each of which will likely be doing various things and fulfilling different purposes. In reality, as we'll discuss later in this chapter, we feel that PEPs may be divided into various categories. Let's take a closer look at this conceptual framework.

A Zero-Trust architecture in concept

We offer a notional Zero Trust architecture in *figure 3.3*, which depicts the security and IT infrastructure of our sample firm, but from a Zero Trust perspective:

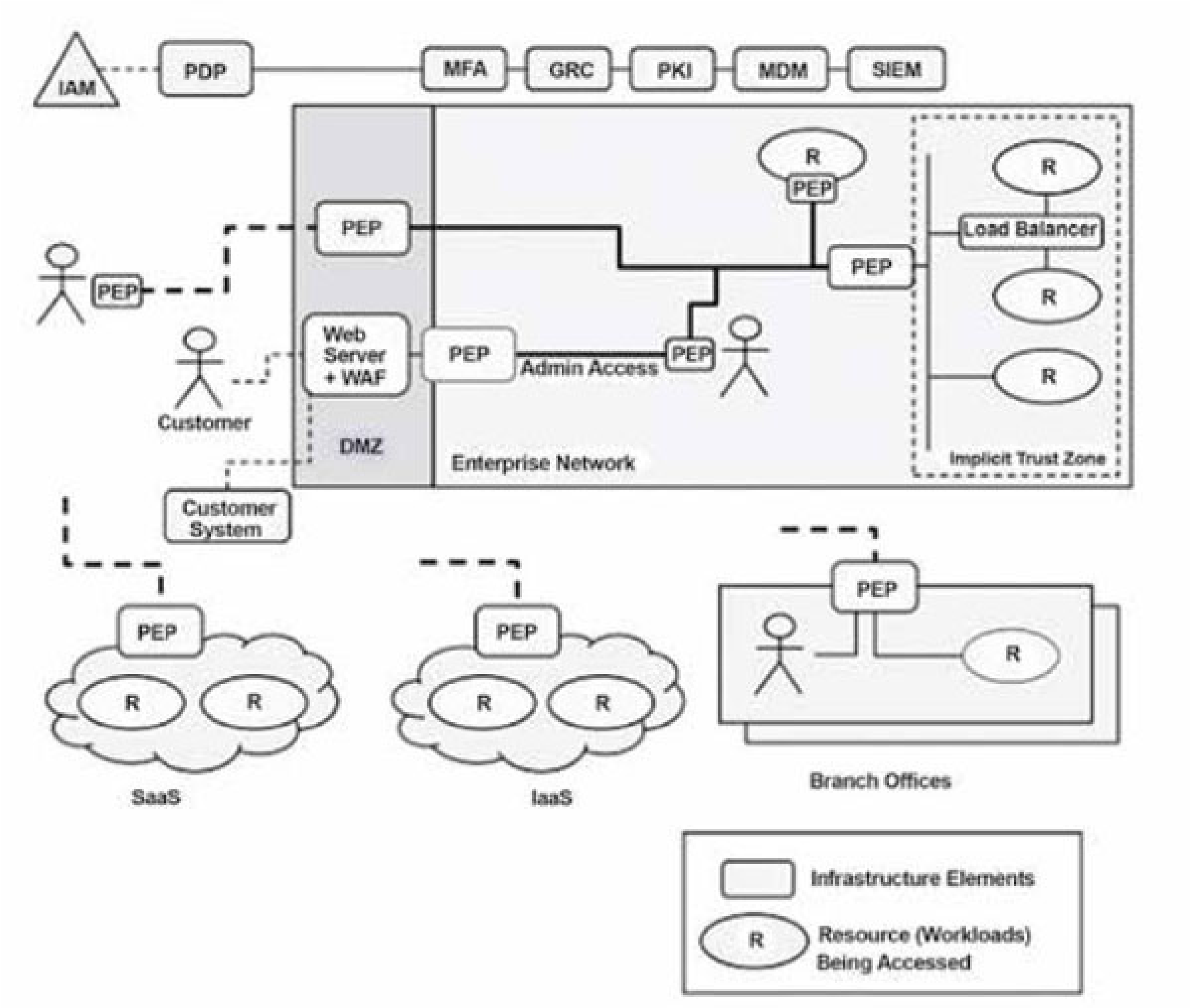


Figure 3.3: *Conceptual Zero Trust Architectur*

The first thing to notice is that any Zero Trust system has a logically centralized Policy Decision Point at its center. In fact, the PDP will most likely be a collection of heterogeneous technological systems glued together by integrations and business processes in a real-world enterprise system evolving to enable Zero Trust.

Of course, Zero Trust is primarily an identity-centric system, and any PDP must have a close, trusting connection with the organization's identity suppliers. The PDP may have a direct network connection with the IAM provider (for example, if it uses LDAP or RADIUS) or an indirect connection (for example, if it uses SAML).

What's more crucial is that the PDP must be set such that it can trust the data it gets from the identity provider, whether directly or indirectly. This is often accomplished by providing the PDP with a service account that allows it to perform API calls to the identity provider or by providing the PDP with the identity system's public certificate, which allows it to validate the data.

Second, since they are employed inside the PDP's policy model, the PDP must be able to transfer identity attributes from the identity provider (or providers) into its internal representation. We'll go over the policy model in detail in *Chapter 17: A Policy of Zero Trust*, but we'll give you a quick overview now.

The NIST paper is a nice place to start: *Access to resources is decided by dynamic policy—including the observable state of client identification, application, and the requesting asset—and may include additional behavioral variables*, it claims as one of the key Zero Trust concepts. To put it another way, if a subject has access to a resource, there must be a policy in place that has been examined and currently permits the subject access to the resource.

Enterprise resources should not be accessed without accessing a PEP, according to the NIST Zero Trust guideline, which is why PEPs are spread across the enterprise architecture in *figure 3.3*. Note that the PEPs are located at different locations and serve distinct roles in our diagram.

In reality, as we'll discuss later in this chapter, we feel that PEPs may be divided into various categories. Let's take a closer look at this conceptual framework. A successful Zero Trust solution, in our opinion, will include a collection of PEPs that are centrally maintained while being spread throughout the company ecosystem. The system is in place. PEP behavior must be controlled by a set of dynamic, context-sensitive regulations that are implemented throughout the environment. However, as previously stated, these PEPs come in a variety of shapes and sizes, each with its own set of duties and functions. The PEP stationed in the DMZ, for example, is responsible for allowing only certain types of traffic users who have been approved and authenticated and have access to the right set of internal resources. It must do so based on the set of rules, which must be enforced at the network layer. Permissions granted to it by the PDP, which the PDP gets from policies user and system context are among the different inputs. By looking at the mechanics, we'll be able to learn more about them.

This is something that can happen later in the book. Another example: the PEP executing within the Resource in the diagram's upper right must enforce the rights granted to it by the PDP. This PEP may be in charge of enforcing role-based permissions within the application, or it may be in charge of regulating inbound (and presumably outgoing) network traffic.

The PEPs receive their orders—the policies that they are responsible for enforcing—from the PDP in both circumstances. In *Chapter 17*: *A Policy of Zero Trust*, we'll go into policies in greater detail, but first let's set the stage. We're going more detailed than NIST because we feel this extra structure is valuable and will give you with a helpful framework for thinking about, designing, defining requirements, selecting solutions, and eventually deploying a Zero Trust platform in your organization.

Policy components

A policy is defined as a declarative declaration that states that a subject may conduct an action on a target if and only if specific criteria are satisfied. The *table 3.1* depicts this:

Components	Description
Subject	The beings executing (initiating) acts are referred to as subjects.
Criteria	Topics must have verified identities, and policies must provide subject criteria that identify which subjects are covered by the policy.
Action	The activity that the person is engaged in. Either a network or an application component is required, and both may be present.
Target	The resource (object) on which the action is being done. This can be defined statically or dynamically inside the policy, and its scope can be vast or small, but narrow is recommended.
Condition	The conditions in which the subject is allowed to carry out the action on the target. The Zero Trust system must allow for the creation of conditions based on a variety of factors, such as subject, environment, and target properties.

Table 3.1: *Policy Components*

Let's have a look at the policy structure. The collection of identities (subjects) to whom this policy applies is defined by subject criteria. Subjects, who might be individuals or **Non-Person Entities** (**NPEs**), must be validated and be part of an identity management system. Many qualities are connected with subjects, and they are derived from a variety of sources including their authenticating identification system, device profile, and network or geolocation information, among others. These qualities are used in the subject criteria to assess whether or not a policy should be assigned to a certain identity (notice that attributes are also used in conditions, which we'll cover momentarily). Even from this little introduction, it should be evident that a Zero Trust policy model applies **Attribute-Based Access Control** (**ABAC**) in a variety of ways. The actual activity that this policy authorizes the subject to do is defined by actions. It can have either a network or an application action, but not both.

The system or component that is being operated on is referred to as a target. These can be specified statically (for example, with a set hostname or IP address) or dynamically (for example, with hypervisor or IaaS labels or tags that are resolved at runtime). They might be very specific (for example, a single service operating on a single server) or very broad (for example, access to a group of servers or a subnet). Conditions govern when the subject is permitted to conduct the action on the target, and they might encompass a wide range of conditions.

Let's use an example to help you visualize how the PDP interprets them and how different sorts of PEPs could operate. A sample policy for managing access to an internal web application is shown in *table 3.2*. In *Chapter 17: A Policy of Zero Trust*, we'll go through this and other cases in detail:

Billing department users must be able to access the Billing web application.	
Subject criteria	Users who are members of the Identity Provider's Dept Billing group.
Action	Users must be able to access the Web UI over HTTPS on port 443.
Target	billing.internal.company.com is the FQDN for the billing application.
Condition	On-premises or remote users are also possible. Prior to access, remote users must be requested for MFA (at the time of authentication). Users must access this application through a company-managed device that is protected by endpoint security software.

Table 3.2: *A sample policy*

Policy enforcement points: What are they and how do they work?

Let's take a closer look at the PEPs now that we've covered the basics of policies. As previously stated, policy enforcement is carried out by PEPs at various levels and kinds, as shown in *figure 3.4*:

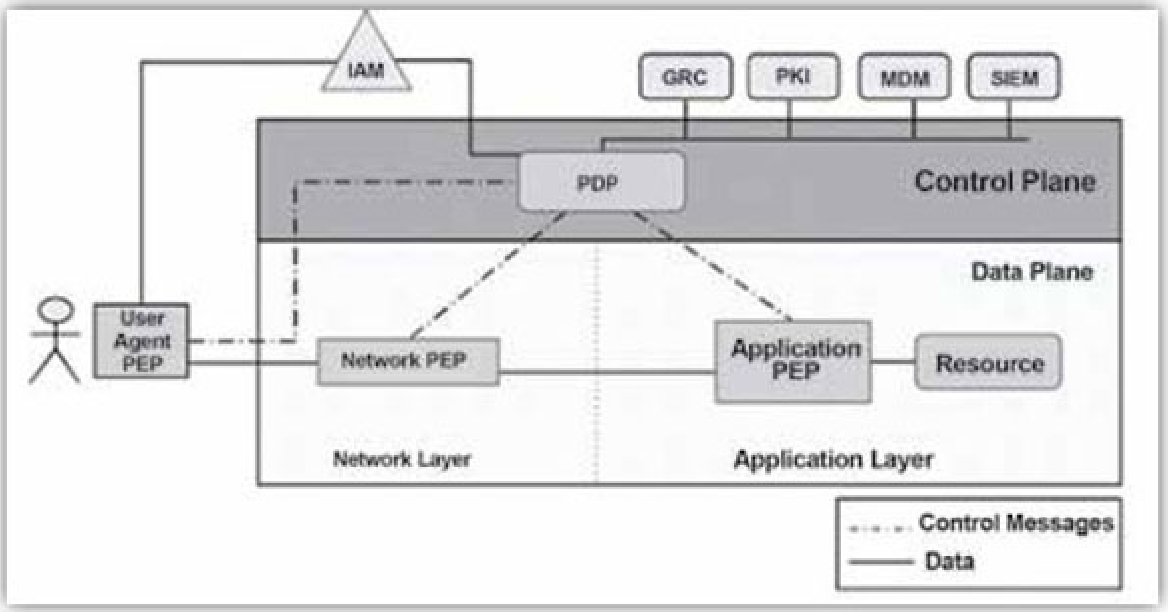


Figure 3.4: *Control plane, data plane, and policy enforcement layers*

As shown in this diagram, we believe there are three categories of PEPs: user agent PEPs, network PEPs, and application PEPs. Because Zero Trust networking is the most popular starting point and is mainly the direction of the NIST publication, network PEPs are likely the easiest conceptually in Zero Trust models.

Many businesses already have network PEPs in place—to some extent, corporate firewalls (NGFWs) may be considered Zero Trust PEPs, but with certain drawbacks that we'll cover later. These PEPs can undertake inline enforcement of network traffic since they operate at the network layer, which is why they are natural enforcement points. They may also inspect traffic, whether it is metadata or actual traffic data.

External application PEPs (such as a PAM or DLP system) and internal application PEPs (such as an agent operating on a workload) are both possible. The PEP can be used to enforce policies locally on the host, such as local OS firewall rules, in the latter scenario. Furthermore, the PEP may potentially be part of the programme itself, relying on external characteristics or actions to influence it. This is a critical feature: PEPs must have some level of integration with the PDP and be able to enforce aspects of the policies that the PDP provides. This enforcement might be limited to the application itself (for example, ensuring that a given identity has an account with a certain application role). Modern apps, for example, that offer just-in-time provisioning depending on the contents of a SAML assertion, are instances of this. This provisioning can take the shape of a user's roles being changed or a new account being created with an initial role. User agent PEPs are components that operate on a user's device and provide services that are frequently required for Zero Trust systems, such as establishing an encrypted connection across an untrusted network (this is referred to as *coordinating the connections* by NIST). These PEPs are frequently used to inspect a device in order to acquire information that is then utilized as input into policies (for example, device configuration and security posture). The PEP can also communicate with the subject (end user), for example, by requesting extra authentication or informing them. While this PEP is not required, many (if not all) commercial Zero Trust systems include a user agent (client) that may be installed on user devices. Most commercial Zero Trust systems also provide clientless or web-based access options, albeit these often have limited capability. We'll show a user agent PEP in place in all of the figures in this book.

It's worth noting that there's a blurry border between various sorts of PEPs in some circumstances, and their roles may overlap. IDS/IPS, for example, can be network-based or host-based, as our industry is well recognized.

DLP functionalities can also be implemented on a host or within a network device, such as an NGFW. It is unimportant if certain enforcement points, such as DLP and PAM, function at the network or application layer (or both). What matters is that both DLP and PAM should be considered Zero Trust PEP components, and their rules should logically be included in the Zero Trust paradigm. To drive this, there should be synergy between them and the Zero Trust system. It might be triggered by identity attributes/roles or by a different Zero Trust policy model, depending on how it's implemented.

In the end, the platform you pick and how you deploy it will determine the functionality and behavior of your PEPs. The core of what we're doing throughout this book is showing how your present infrastructure and architecture should be viewed as a series of Zero Trust PEPs. All of your PEPs must be integrated, share a policy model, and be operationally connected for your Zero Trust journey to be successful.

However, bear in mind that this is a goal, not a starting point, and that many old infrastructure parts will still be in existence that aren't PEPs and aren't logically tied into your Zero Trust policy model. The *figure 3.3*, for example, still shows a load balancer, which serves a beneficial purpose even under the Zero Trust design. The load balancer is functioning at a low-level network level in this situation, and it may continue to function normally despite the adoption of Zero Trust across the rest of the architecture.

It doesn't need to be extremely complicated, and its operation isn't affected by user or system context. This will be true for many aspects of your infrastructure, so although Zero Trust can and should prompt you to reconsider your security and integration design, it does not necessitate a complete overhaul.

To put it another way, it's possible to gradually implement Zero Trust and begin enforcing regulations at crucial places throughout your infrastructure while avoiding major changes. This brings us to the second point: policies.

What is a policy enforcement point, and how does it work?

Policies are at the heart of any Zero Trust system, and PDPs and PEPs are constantly evaluating and enforcing them. However, this presents an intriguing and philosophical question: what qualifies a security component as a Zero Trust Policy Enforcement Point? Can a 5-year-old simple firewall, for example, be called a PEP?

The answer is *it depends*, as it does with most fascinating issues, and understanding comes from the thinking processes involved in evaluating the dependencies, so let's get started.

Of fact, our simple firewall is a *network enforcement point* in the sense that it enforces access control rules like *Allow TCP traffic on port 443 from source subnet 10.5.0.0/16 to reach destination subnet 10.3.0.0/16.* We contend, however, that this firewall is not a Zero Trust PEP since it fails to fulfil the following criteria:

- Be able to enforce the identity-centric and context-sensitive policy paradigm of the PDPs
- Automatically respond to PDP-driven policy changes.
- When communicating with the PDP, use a control channel.

A traditional firewall, as we'll see later in the book, cannot match these needs; in fact, as we'll see later in the book, the capacity for a PEP to be programmatically driven by the PDP and to update its rules in an automated manner is a vital element in achieving Zero Trust. That is, our core concept is that a Zero Trust system must be capable of enforcing dynamic regulations based on identity and context. As a result, every PEP must be able to receive continuous updates from the PDP and automatically alter the policies it is enforcing in near-real time and without the need for human interaction. Even at a small scale, this is the only way to accomplish the responsive, dynamic character of Zero Trust.

So, let's continue this mental exercise. What if our 5-year-old firewall, which is dusty and lying in a wire closet, gets a policy-driven automation layer installed on top of it? In this situation, we'd argue that this firewall, when combined with network security automation software, might now be deemed a Zero Trust PEP, as long as the network security automation solution is also integrated into the PDP and fits the criteria outlined above. The key of a Zero Trust PEP, in other words, is that it has automatic interaction with the PDP and can react fast to policy changes.

From a policy model or operational standpoint, Zero Trust Policy Enforcement Points cannot be compartmentalized.

It's worth noting that we used the term *automated* in this case. It's entirely good to have manual phases in the process, such as business process approval for specific changes or manual approval in unusual *breakglass* scenarios. However, for the day-to-day (or hour-to-hour, or even minute-to-minute) changes that this enforcement point regulates, there must be automated modifications.

Consider the Policy in *table 3.2* as an example. Consider what occurs when the Dept Billing directory group adds user Jane. She must be able to access the Billing app at the network layer and have an active account at the application layer shortly after that.

Although creating Jane's account may be a laborious procedure in reality, we feel that network access modifications must be automated. Consider what happens if Jane unintentionally opens a phishing link and her laptop is infected with malware that begins doing network reconnaissance. To prevent the virus from possibly compromising the business-critical billing system, the enterprise's security systems identify this as a sign of compromise and respond by immediately restricting her network-level access to it. The network PEP must respond immediately and autonomously; it cannot wait for a business process to complete. Note that in our situation, the network PEP is extremely likely to prohibit this access solely at the network layer. There's no need to update anything in PEP; this is just a temporary problem with Jane's laptop. In reality, a well-designed Zero Trust solution would allow Jane to access the billing application from another device (such as a desktop computer) while blocking the laptop. In *Chapters 5: Identity and Access Management (IAM)* and *Chapter 17: A Policy of Zero Trust*, we'll go further into these themes.

Models of Zero Trust deployment

Following that, we'll look at a few Zero Trust deployment models, including two from the NIST Zero Trust specification and two more for good measure. These models offer us with a higher degree of detail regarding how Zero Trust systems may be implemented in the real world, while real-world deployment designs will of course be based on the capabilities of the selected technology. Many of the vendor-provided enterprise Zero Trust approaches, we believe, will align with one or more of the deployment methods outlined below. In other words, these deployment models will serve as a good foundation for evaluating possible providers and weighing their benefits and drawbacks. These aren't meant to be exhaustive, but they should serve as a good starting point. They're also not always mutually exclusive; some systems may incorporate components from many paradigms. Note that in the following talks, we'll concentrate on the differences between these models rather than what they have in common.

***Finally, notice** that we've omitted the PDP's link to identity management and other business security systems from the following diagrams for clarity. Regardless of the Zero Trust deployment approach you pick, those connections must still exist.*

Deployment model based on resources

***The resource-based** deployment model (shown in [figure 3.5](#)) is the first approach we'll look at:*

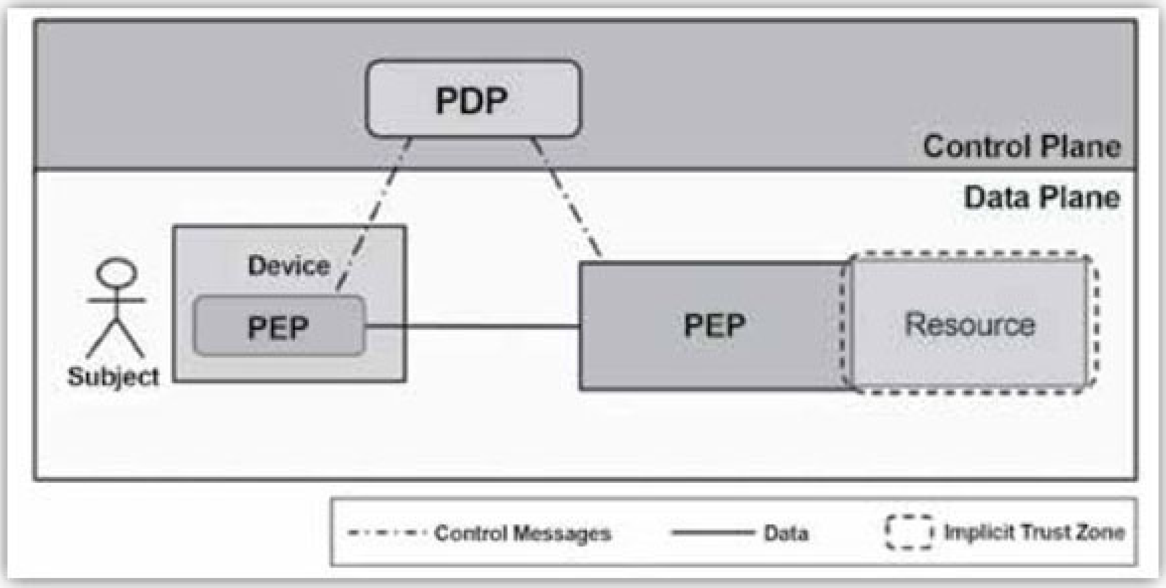


***Figure 3.5:** Resource-Based Deployment Model*

What's crucial about this paradigm is that, first and foremost, a user agent, known as the user agent PEP, is usually installed on the subject's PC. 8 Second, there's an inline PEP (the gateway) that's installed *on the resource or as a component immediately in front of a resource*, according to NIST (emphasis ours).

This figure also depicts the implicit trust zone, which is a space behind a specified PEP in which all resources (entities) are trusted to the same degree. This reflects the security domain's border for which that PEP is accountable. Any interactions between components that remain within the implicit trust zone are, by definition, beyond of the PEP's control.

If the PEP is executing on the local resource Operating System, the implicit trust zone consists of the collection of local processes and their interactions within the local OS in the prior example. Naturally, you want to keep the implicit trust zone as small as possible, while keeping in mind that each deployment strategy comes with its own set of trade-offs.

Pros of the resource-based deployment model include:

- End-to-end management of application access and network traffic
- A small implicit trust zone "*behind*" the gateway

All network connections between user devices and the target resource are encrypted, and access control restrictions are enforced, according to this approach. It also guarantees that the PEP (and hence the organization's Zero Trust security approach) enforces all network communication with the resource. This paradigm, however, has a number of drawbacks that must be addressed.

Cons of a resource-based deployment model:

- PEPs must be installed on both user devices and resources.
- Technical clashes between resource components and PEPs are a possibility.
- PEPs must be able to run on a broad range of operating systems, some of which may be obsolete or legacy.
- There's a chance that application resource owners will object.
- A 1:1 link between PEPs and Resources is required.
- Using an end-to-end secure tunnel can make inline security safeguards ineffective.
- Remote users must be able to see and utilize PEP.

First, this strategy necessitates the deployment of a PEP on every resource in the environment, which might be troublesome. Any environment larger than a small one would almost certainly necessitate a high level of automation, especially in virtualized or cloud settings. Locally deployed PEPs may also cause technical issues inside the same operating system, such as with components that manage network or disc I/O, such as web servers or databases.

This architecture also necessitates the deployment of PEPs on all protected resources. This is frequently a multi-faceted difficulty. First and foremost, the PEP software must be supported and deployable across all workloads. Many businesses have legacy programmes that operate on mainframes or minicomputers and are unlikely to support a PEP.

These older programmes, however, are frequently the ones that require the greatest attention in terms of security! Second, many security teams may face resistance from application owners who are hesitant to integrate any new software to their revenue-generating or mission-critical programmes.

From an operational standpoint, this strategy necessitates the deployment of one PEP for each controlled resource, putting a significant administration burden on the Zero Trust system and the team in charge. If a virtualized or cloud-based system is used, for example, many ephemeral workloads, frequent onboarding, and other factors make up the environment.

Offboarding these resources should be avoided at all costs. You'd have to do something like this in this scenario. Ascertain that the Zero Trust mechanism is sufficiently automated to handle this churn. This architecture ensures that network traffic is encrypted from the start as a basic Zero Trust tenet. the resource PEP to the user agent PEP This is the case in many commercial Zero Trust solutions. An encrypted tunnel is used to do this. This is safe and effective, but it usually comes with some drawbacks. As a result, any intermediate will be unable to see all of that traffic. This is advantageous. If the intermediary is an attacker, it's beneficial, but if it's an enterprise-wide security system, it's harmful component, such as a network-based IDS/IPS. Finally, and maybe most crucially, the PEP enforcing resource protection must, Of course, subjects, including distant users, must be accessible. As a result of the PEP in this When a model is included in a resource, it means that all of the subjects are on the same page. Each PEP is connected to the same physical network, or all PEPs are immediately accessible from a distant location. The very first is the first option is unlikely to be true, and the second alternative is unlikely to be practicable, given the circumstances a large number of these resources are located on private network segments.

Deployment model based on enclaves

The enclave-based deployment approach, shown in *figure 3.6*, is the second option. The PEP is in front of numerous resources in this situation, which is referred to as a resource enclave. This group of resources might be geographically close (for example, in a collocated or on-premises data center) or logically connected (for example, a set of cloud-based or virtualized servers).

The subject, like the previous paradigm, has a locally installed user agent PEP:

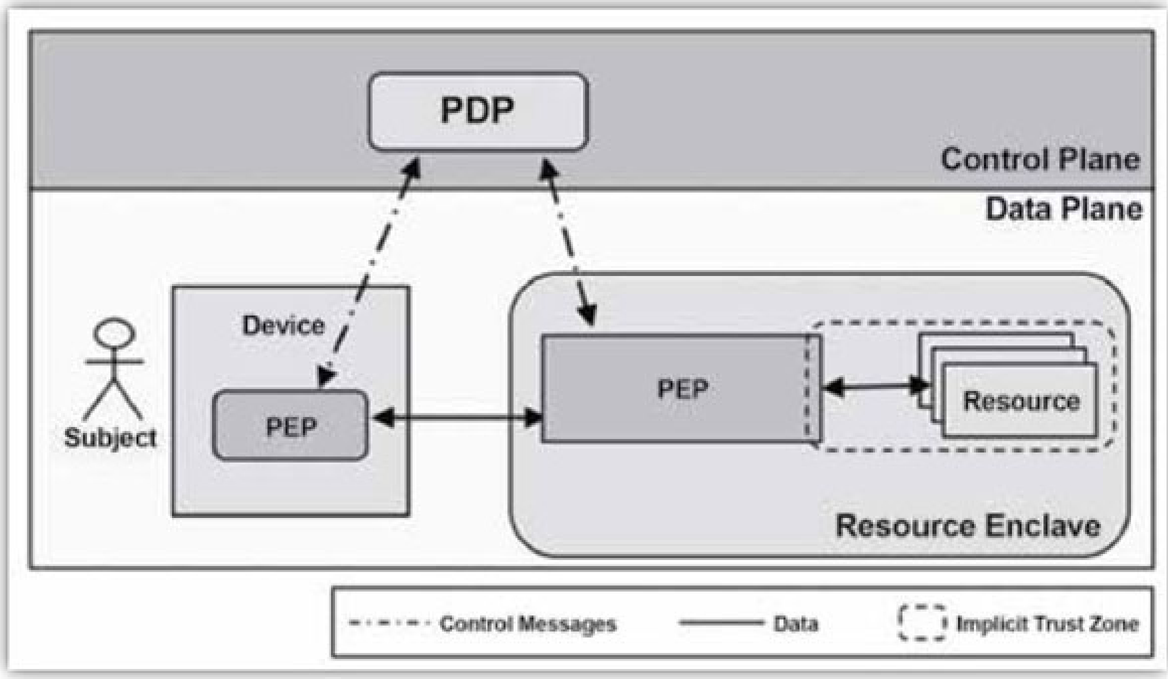


Figure 3.6: *Enclave-Based Deployment Model*

What's vital to remember is that the implicit trust zone in this model involves many networked resources that are most likely talking with one another. That is, under this paradigm, the resource enclave must run only on a logical private network that is under the authority of the organization. We use the word *logical* because, while this might be running in a public IaaS or shared colocation environment, the network traffic at layers three and higher must be private to the organization.

Although resources inside the enclave can and do interact with one another outside of the PEP's visibility and control, the only route for subjects outside the trust zone to communicate into it is through the PEP, which is governed by policy.

That is, using this paradigm, businesses must ensure that they fully comprehend the data and communication patterns of the resources. It's also worth noting that this Zero Trust deployment methodology favors a *user-to-service* approach.

Model of Deployment Based on Enclaves: Pros

- Fewer PEPs installed.
- Handles ephemeral workloads and dynamic situations effectively.
- PEPs can run at the network's edge (DMZ), serving as natural ingress points.

The one-to-many link between PEPs and resources makes this model easier to deploy than the previous one because there is an order of magnitude less PEPs deployed. Eliminating the need to put any extra software on the resources not only simplifies operations, but it also eliminates the majority of technical and political issues with application owners. It also has the benefit of installing PEPs at the company network's edge, allowing them to act as a natural ingress point for distant users. Of course, they'll also act as a Policy Enforcement Point for local users, whose traffic will stay entirely within the organization.

This paradigm may be able to handle ephemeral or dynamic workloads depending on how the PEPs are implemented. The goal is for the PEPs to be able to respond to changes in the protected resources, such as recognizing the creation of new resources and applying rules to them based on resource properties (metadata). A PEP guarding an on-premises virtual environment, for example, may receive an API request from the hypervisor signaling the creation of a new instance.

The PEP may instantly apply the relevant policy and provide access to just the permitted set of users based on the attributes of this instance. The same function may be performed by a PEP operating in an enterprise's IaaS environment.

Cons of an Enclave-Based Deployment Model

- PEPs are a new sort of entry point into the company network, with the potential to be huge, opaque, or noisy implicit trust zones.

The most difficult aspect of this paradigm is determining the size and breadth of the implicit trust zone, which will, of course, be determined by how and where you deploy these PEPs. This paradigm is a great basis for Zero Trust because it combines a concentrated collection of resources with well-understood and well-managed communication pathways.

This concept is especially well suited to organizations that operate in modern (primarily IaaS-based) systems or use programmatically driven infrastructure (such as DevOps).

In order to limit the size and breadth of each implicit trust zone, organizations with lower operational maturity, lesser visibility, or complicated legacy networks may need to install additional PEPs. Alternatively, they might use a hybrid strategy offered by some Zero Trust suppliers, which combines this concept with the micro segmentation paradigm we'll talk about later.

Another disadvantage of this strategy is that it is typically more political than technical.

The PEP is often deployed in an organization's DMZ, at the corporate network's edge, under this approach. That is, it is designed to be accessed via the Internet, which is the least trustworthy location in the known cosmos. This is required for distant users to have access to protected resources, but it also provides a possible attack vector, similar to a VPN concentrator. New edge devices should be evaluated and scrutinized by security and networking teams, but they occasionally resist for non-technical reasons. It's critical for Zero Trust teams to be aware of this, as well as to get adequate management support for their project, so that these new edge devices can be appraised fairly and objectively. As an aside, certain edge PEPs actually provide greater network security than typical edge devices, so forward-thinking networking and security teams should welcome the move.

Model for Cloud-Routed Deployment

All traffic from the topic is routed through a cloud environment before accessing the resource in this paradigm, hence the name *cloud-routed*. This is a widely used concept, with several commercial vendors offering it as a service. The *figure 3.7* illustrates this concept:

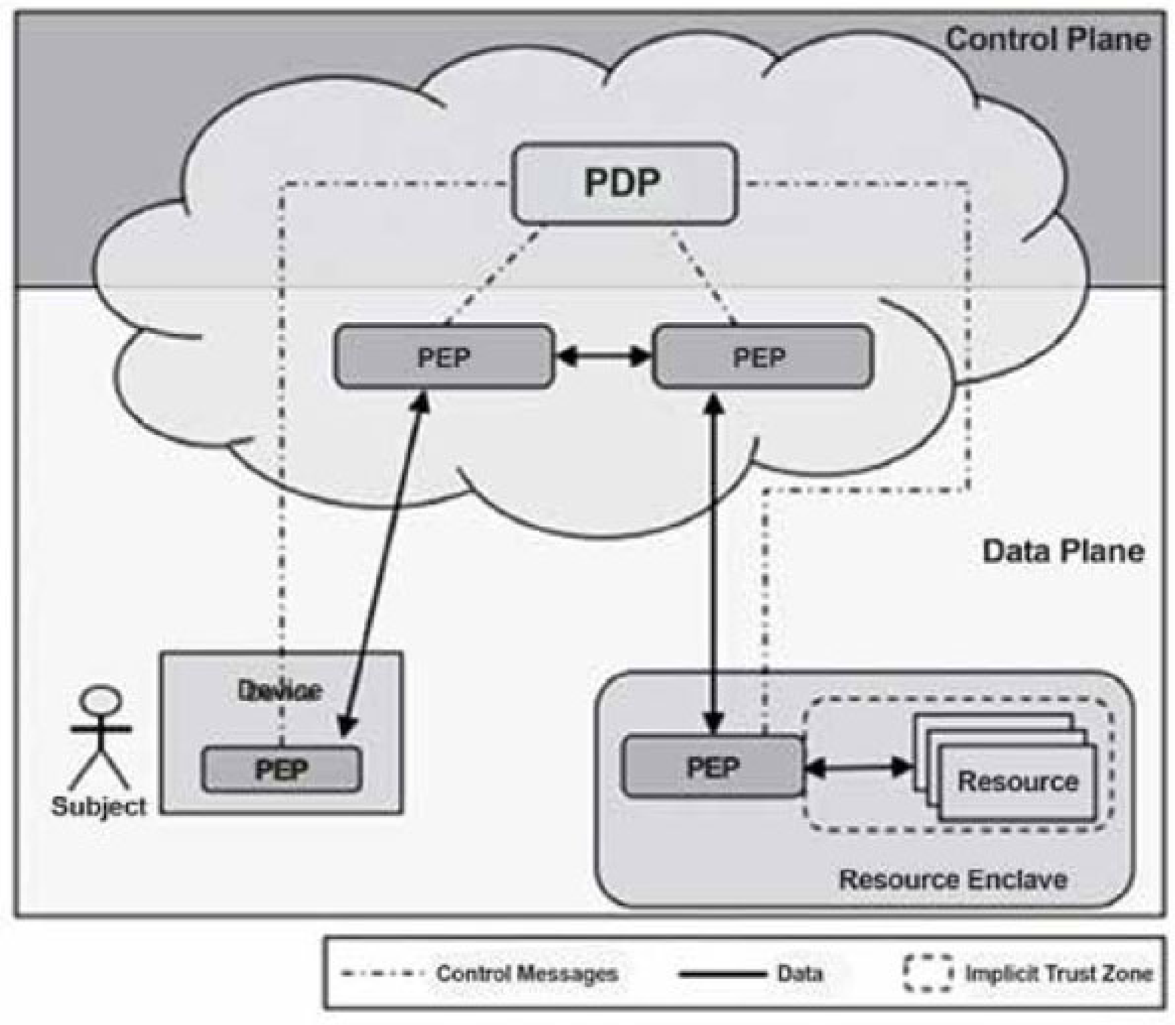


Figure 3.7: *Cloud-Routed Deployment model*

The PEPs in front of the enterprise's resource enclaves in this architecture behave similarly to the PEPs in the previous paradigm. These PEPs, however, vary in one essential way: they do not function as an ingress point into the workplace network. Instead, that responsibility has been legitimately delegated to the vendor's cloud-based PEPs. The enterprise-based PEPs operate as connectors in this approach, creating outbound connections to the cloud-based PEP. Because these on-premises connectors do not require any inbound connections, they typically make this model's adoption easier—at the cost of several constraints, which are explained further below. When a subject wishes to connect with a resource, they must first authenticate with the PDP, after which their traffic is routed to one of the cloud-based PEPs, usually the one nearest to them in terms of geolocation (or perhaps exhibits the lowest latency). Their traffic is then routed through the cloud's PEPs to the PEP that is connected to the target resource enclave. In the same way that the previous model did, the on-premises PEP secures a resource enclave.

Pros of a cloud-routed deployment model:

- Easier to set up for businesses.
- The enterprise's operational overhead is reduced by using an As-a-Service platform. A **Secure Web Gateway** (**SWG**) service is offered by certain providers that use this approach.

Because the on-premises PEPs in this paradigm only make outbound connections, they're usually quite simple to set up. Because they don't need any modifications to DMZ firewall rules or the deployment of any software inside the DMZ, they can evade inspection from network and compliance teams. These PEPs may be installed anywhere within an organization and offer remote access to the network. While this might be a benefit, it could also be a drawback. This technological capacity must not be used as an excuse or a tool to circumvent security, network, or GRC supervision.

If used as *shadow IT*, this might pose a serious security risk to the firm. Of course, security teams must develop a suitable set of policies and apply the concept of least privilege even after they have been authorized. Poor security measures cannot be excused by ease of deployment.

Finally, some providers that employ this approach pair it with a SWG service to provide secure access to publicly available websites for users. For certain businesses, this combination may be desirable since it simplifies implementation and operations.

Cons of a cloud-routed deployment model:

- Without sufficient security, network, or compliance supervision, PEPs can be implemented.
- Increases user traffic delay.
- Typically, just a few network protocols are supported.
- On-premises users accessing on-premises resources are not supported.

- Implicit trust zone that might be vast, opaque, or noisy.

In addition to the potential of this becoming a "*Shadow IT*" remote access approach, there are various other drawbacks to this strategy. First, all user traffic must pass via the vendor's cloud, which adds delay and may lower throughput.

This can be a significant hindrance in particular use cases and applications. Before deploying this to production users, you should have a thorough grasp of vendor platform network performance and conduct some testing. Second, cloud-routed models typically only handle a few numbers of network protocols, such as TCP/IP (and, in certain situations, a few application protocols like HTTPS, SSH, and RDP). This paradigm may not be suitable if your users and applications require alternative protocols, such as UDP, or require server-initiated connections to users. This approach, like the enclave-based model previously mentioned, has the same concerns concerning the implicit trust zone.

Most crucially, because all traffic must pass via the vendor cloud, this approach is best suited for remote users. If users are on-premises and using on-premises resources, their traffic must be "*hair pinned*" through the vendor cloud, which adds delay, reduces throughput, and increases bandwidth use and expenses for the company.

Deployment model for microsegmentation

The last deployment approach, known as microsegmentation, is focused on the server-to-server use case. As the name indicates, this paradigm tackles the problem from the standpoint of resources rather than consumers. In reality, as shown in *figure 3.8*, the resources are the principal subjects (**Non-Person Entities** (**NPEs**)) for which policies must be developed and implemented. For completeness' sake, and since many commercially available solutions enable them, we display a human subject as well, but they are often of secondary relevance in this model:

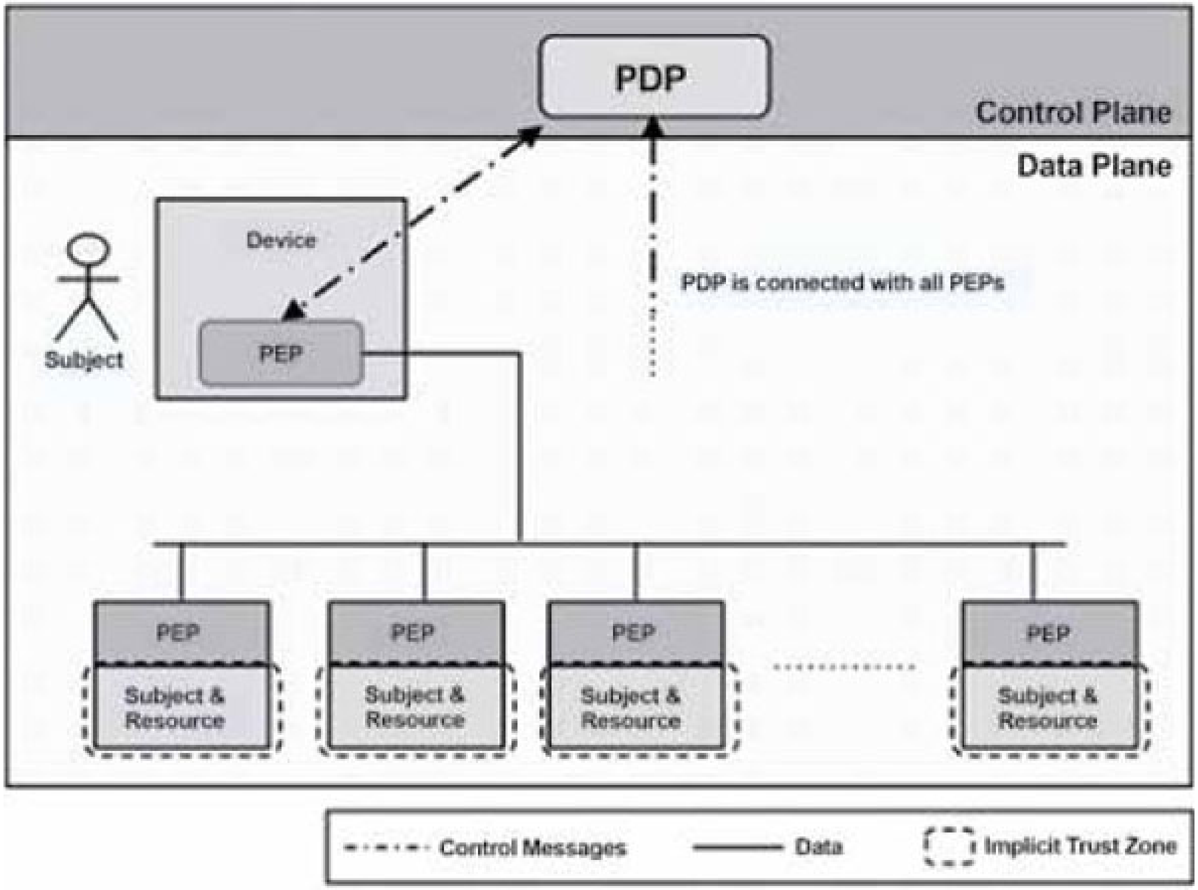


Figure 3.8: *Microsegmentation Deployment Model*

This paradigm is a variation of the resource-based model we described before, with the crucial difference that the resources are now also subjects (authenticated identities). This has important consequences for the policy model and PEP enforcement capabilities, as well as the resource finding and visualization tools often provided by commercial implementations.

In general, NPE subjects' identities will be weaker than human subjects'—often reliant on certificates and clearly based on a single authentication element. This certificate is usually created and managed by the company's certificate authority (PKI).

Pros and cons of microsegmentation:

- Precise, bidirectional resource access control (for servers or microservices)
- Small implicit trust zone

This strategy, like the first, has an implicit trust zone that is normally limited to the resource itself. As a result, it may regulate resource access at a finer level and implement bidirectional regulations. Because the PEP runs locally on the resource, its policies can regulate both inbound and outgoing network interactions. These restrictions may typically be applied to server resources as well as microservices in commercial deployments.

Microsegmentation has certain drawbacks

- Potential for technical incompatibilities between resource components and PEPs if PEPs are deployed on both user devices and resources.

PEPs must be able to run on a broad range of operating systems, some of which may be obsolete or legacy:

- There's a chance that application resource owners will object.
- A 1:1 link between PEPs and Resources is required.
- It's possible that user-to-resource access isn't the best fit.
- There is no built-in remote access; subjects must have direct access to PEPs.

Because the disadvantages of this strategy are the same as those of the first model—namely, the deployment and administration of PEPs on every resource that requires protection—we won't go over them again here. It also has another potential disadvantage: a commercial (or open source) approach focusing on this area may have functional or architectural flaws in the user-to-service scenario. This may or may not be the case for any given implementation, but it's worth noting.

This may or may not be true for every given implementation, but it should surely be included in your assessment criteria.

Conclusion

Although the core ideas of policy decision points and policy enforcement points have been around for a while, their use inside a Zero Trust security paradigm is relatively new. We urge you to utilize this approach to color to guide your needs and goals, as well as to shape your thinking and architecture. To do this, you must begin thinking about current components in your corporate security architecture as PEPs in your fledgling Zero Trust architecture.

This book is designed to assist you in thinking about them in terms of the functions they do, rather than as separate components that happen to fulfil a set of functions.

This is similar to the adage “people don’t want quarter-inch drill bits; they want quarter-inch holes,” in that it emphasizes the value rather than the means of obtaining it.

To return to security, instead of thinking *I need a firewall*, consider *I need a Policy Enforcement Point that can regulate network traffic, as well as a method to specify that policy throughout my infrastructure.* Alternatively, rather than thinking, *I need to place an IDS here to check my web app traffic for SQL injections*, you could consider, *I need to ensure that the web application traffic is screened for SQL injections before it is handled by the app*.

In my architecture, I have numerous PEPs that might help me achieve this aim.” This mental adjustment should aid you on your path.

We introduced and examined a representative enterprise architecture, discussed a generalized Zero Trust architecture, briefly introduced a policy model, and explored several different Zero Trust deployment models in this chapter, which provided a lot of background information on Zero Trust architectures. We’ll look at three case studies in the following chapter to see how these businesses tackled Zero Trust in practice.

Chapter 4

Zero Trust in Practice

Let's look at some real-world instances of Zero Trust systems now that we've covered the basics of Zero Trust and looked at a few models. Two of them, Google's BeyondCorp and PagerDuty's Zero Trust system, have been publicly detailed and are excellent examples of Zero Trust architectures and systems created internally at two very different companies with very different approaches.

Even if we couldn't install a Zero Trust architecture that matches any of them, we can learn a lot from these case studies.

We'll focus our efforts on contrasting their viewpoints, ambitions, and trade-offs through the prism of the Zero Trust concepts and architectures that we've just presented, because those first two instances have been well-documented. Our third example is a company that achieved Zero Trust using the Software Defined Perimeter design, which will let us evaluate the advantages of that approach. Let's get started with our first case study, an internal Google experiment that is credited with generating most of the industry's interest in Zero Trust.

Google's BeyondCorp

BeyondCorp, Google's internal term for their network security transformation programme, is a tremendous accomplishment that has rightly influenced the industry. Google not only reinvented their internal security architecture and provided network access restrictions for tens of thousands of users, but they also documented it in a series of USENIX; login: papers that began in 2014 and ended in 2018.

These well-written and detailed publications have had a significant impact on the business, and we must give Google credit for championing the Zero

Trust ideals. We recommend that you read the source articles; we're only presenting a summary here. Essentially, Google developed and executed a complicated Zero Trust system on a wide scale over a number of years. They established *a new paradigm that does not rely on a privileged corporate network, as they put it. Instead, independent of a user's network location, access is exclusively based on device and user credentials... Based on device status and user credentials, all access to business resources is completely authenticated, approved, and encrypted*.

As a result of their trip, the corporate network no longer has any inherent trust—all access is allowed based on identity, device, and authentication, all of which are based on reliable underlying device and identity data sources. They effectively replaced inherent network trust with earned device trust—they now have a real Zero Trust network, and all internal apps are accessed through the BeyondCorp system, whether the user is in a Google office or working remotely. They also decided to restrict access to internal apps to managed devices only unmanaged and BYOD devices are not allowed access. It's also worth noting that this effort was only focused on limiting user-to-server access rather than server-to-server access.

These design choices have a number of consequences for the project; for example, it relies heavily on high-quality data on device inventories, therefore they created a complex device inventory database to support it. They use a centralized identity system with SSO to issue short-lived access tokens and rely on corporate-issued certificates stored in each device's **Trusted Platform Module** (**TPM**) as a root of trust. Their identity management system is utilized to store user group and role information, providing identity context to their policy decision points. Their identification system is also linked to HR operations, making it trustworthy and up to date. The *figure 4.1* depicts the BeyondCorp infrastructure components:

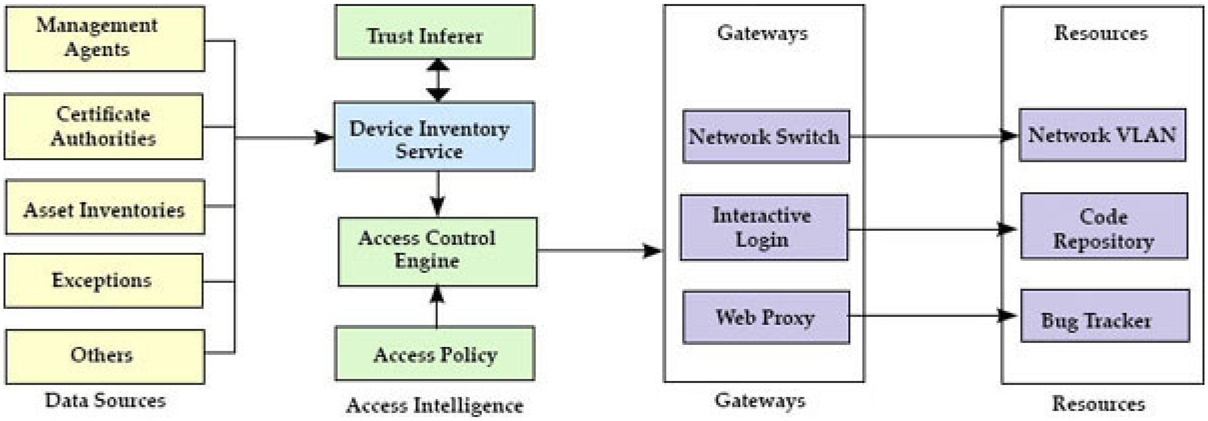


Figure 4.1: *BeyondCorp Infrastructure Components*

The following are the essential components of BeyondCorp. First, the Data Sources (logically) correspond to the external data sources presented in Chapter 3: *Architectures With Zero Trust* Zero Trust models. Naturally, the Resources match to the resources in our model (and are what NIST terms Enterprise Resources). With the other two portions, Google has adopted an unusual hybrid approach. Their Access Intelligence components effectively make up the **Policy Decision Point** (**PDP**), and their Gateways make up the **Policy Enforcement Point** (**PEP**); nevertheless, their Access Control Engine is technically part of their PEP as well. Depending on the application, these resources may also operate as Application PEPs, ensuring fine-grained access. The *figure 4.2*, which combines *figure 4.1* with the Zero Trust architectural components described in Chapter 3*: Architectures With Zero Trust*, depicts this layered perspective.

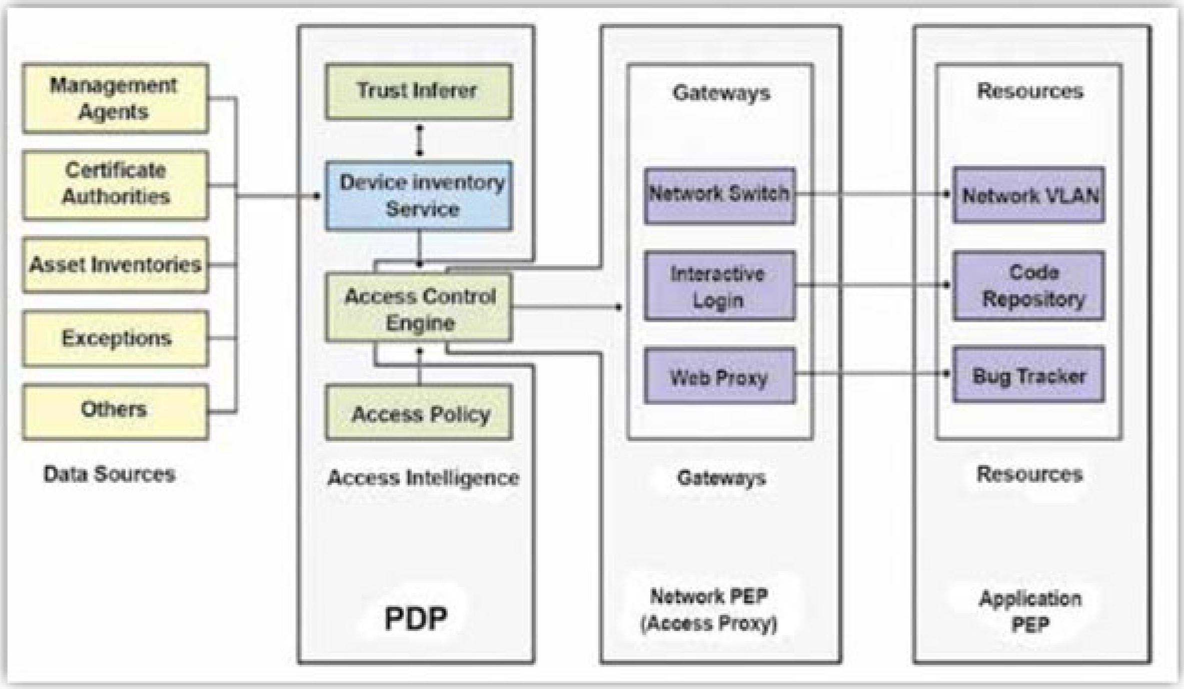


Figure 4.2: *Annotated BeyondCorp Infrastructure Components*

The BeyondCorp Access Proxy (which is composed up of Gateways and is a component of the Access Control Engine) serves as a PEP for both remote and on-premises users. The system establishes a trust level using numerous data sources, with dynamic enforcement within the Access Proxy at the moment of access. This is an excellent example of dynamic behavior that adheres to NIST concepts, such as regulations based on group membership and device properties. The Access Control Engine makes choices on a per-request basis, according to Google's publications. This is one of two situations where the BeyondCorp implementation blurs the borders between some of the components in the Zero Trust architectural model that are conceptually separate (this is common—we'll see some fuzzy lines later in this chapter when we discuss the Software-Defined Perimeter).

Although the Google articles do not clarify to what extent the application PEPs are hooked into the access policy, or to data sources such as IAM, the Access Proxy is described as offering coarse-grained enforcement at the front end, with permission enforced at the back end (inside the resource). To identify managed devices from unmanaged devices, their on-premises Network Access Control system employs dynamic VLAN assignment based on *figure 4.2*. Annotated BeyondCorp Infrastructure Components device certificates. This is a successful technique of integrating their 802.1x-based NAC into their Zero Trust network, despite being quite coarse-grained. BeyondCorp is especially intriguing since it incorporates both the enclave-based and resource-based models. The Access Proxy sends additional security details to the sites being visited through HTTP headers. The benefit of utilizing HTTP headers to convey this metadata is that any resources that aren't expecting it or can't process it will silently disregard it. This technique decreased the time and work necessary to roll this out across hundreds of Google applications, allowing most of them to be onboarded without change while also allowing the use of this data for increased security in some of them. It's worth noting that this method incorporates control messages into the data plane. This isn't *wrong*—it's a clever design choice that makes a lot of sense within the BeyondCorp architecture, and it shows how the notional Zero Trust model can be applied to a variety of implementation designs.

BeyondCorp was a lengthy, complex, and multi-year deployment and organizational shift, according to the Google team. The sheer scope and complexity of Google's organization and network were part of the cause. Another feature of this group was that it was a true pioneer, innovating, learning, making mistakes, and iterating throughout the process. The good news for the rest of the security industry is that they've shared so much about their implementation that we've created an ecosystem of commercial and open-source tools, technologies, platforms, and approaches that help businesses achieve many of the same benefits faster, using more structured, predictable, and repeatable approaches. This leads to the next logical question: Can I use BeyondCorp in my company? "No, and yes," is the solution to this question. BeyondCorp is clearly a Google internal initiative and platform that is not available for licensing or re-use. BeyondCorp is extensively interwoven within Google's business architecture, technological infrastructure, and HR procedures, according to their published papers. So, the answer is "no" to the question *Can I implement the BeyondCorp platform for my organization? Can I implement a security solution that gives similar benefits to BeyondCorp for my organization?* is a better question. which has a booming "yes" as a response.

This leads to the next obvious question: Can my firm utilize BeyondCorp? The answer to this question is "no, and yes." BeyondCorp is obviously a

Google proprietary project and platform that cannot be licensed or reused. According to their published papers, BeyondCorp is deeply embedded in Google's corporate architecture, technology infrastructure, and HR procedures. So, to the question, "Can I adopt the BeyondCorp platform for my organization?" the answer is "no." "Can I develop a security solution for my organization that provides similar benefits to BeyondCorp?" is a better question, to which the answer is a resounding "absolutely."

The Zero Trust Network of pagerduty

In comparison to the BeyondCorp example, the PagerDuty case study, which was initially published in the well-regarded, shows a strong contrast. PagerDuty's network is first and primarily focused on safeguarding server-to-server access, whereas BeyondCorp's network is focused on the user-to-server situation. Second, instead of safeguarding access to resources on a corporate network, PagerDuty needs to secure access to resources across different public cloud environments.

Because these diverse cloud platforms offered a broad range of security capabilities (from good to poor), their Zero Trust solution acted as a normalization layer, simplifying things. We've seen Zero Trust solutions in companies have a similar beneficial impact, simplifying operations and setup across numerous heterogeneous and hybrid settings using a common policy architecture.

To automate and operate their virtual servers, PagerDuty's system is significantly dependant on their configuration management system, which was in existence prior to their Zero Trust mission. This was a critical basis for them since it functioned as a "source of truth" for all of their resources as well as a platform for automation. This is, in effect, a hybrid of the Policy Decision Point and the Control Channel.

What's intriguing about this is how it compares to BeyondCorp, where the source of truth is a corporation.

It was a combination of their stringent device control system and their identity management technologies that made this possible. In order to have an authoritative resource catalogue, server-to-server Zero Trust systems often require a robust Configuration Management Database (or rely on network discovery capabilities). User-to-server systems, on the other hand, often rely on identity management as their authoritative mechanisms.

Based on their configuration management and automation technology, PagerDuty's approach leverages a central PDP. They have a distributed collection of PEPs that leverage local iptables firewall rules on hosts to provide a consistent enforcement method across many cloud settings. This method should be familiar to you—essentially, it's the microsegmentation deployment methodology described in *Chapter 3: Architectures With Zero Trust*. In this situation, the PEPs are the built-in host-based local firewalls, which are controlled via their configuration system (the PDP). In order to provide network privacy, their technology leverages a mesh of IPsec connections between all servers on their network.

PagerDuty's methodology and design, by all accounts, performed effectively, however there were a few snags, as with any freshly created and sophisticated system. They haven't published much information regarding their policy approach, but in essence, they assign each server a role that controls access rules, and all servers in that role have the same configuration. This strategy makes sense in a server-to-server environment since servers are often installed in fixed locations and are completely under the authority of the organization. That is, a well-run system will have total control over each server's image, configuration, and network, especially if it is controlled by an automated configuration system like Chef.

User devices, on the other hand, are usually transportable, run-on untrusted networks and in untrusted surroundings and are frequently a "*wild west*" of arbitrary and unique configurations. (User-to-server access management is made much more difficult by BYOD.)

We applaud PagerDuty for their creativity, and we appreciate Evan and Doug sharing it in their book. Given their varied focus on the server-to-server use case, this was a successful project for PagerDuty, and an intriguing contrast in design decisions and issue space when compared to BeyondCorp. We particularly appreciate how they define policies based on data from their configuration management system, which is read in, assessed, and transformed into firewall rulesets enforced by PDPs.

A popular (and recommended) pattern is policies that employ target resource metadata as input, which we go over in detail in Chapter 17: *A Policy of Zero Trust*.

Zero Trust and the Software-Defined Perimeter

The **Software-Defined Perimeter** (**SDP**) is an open security architecture that was first released by the Cloud Security Alliance in 2014 and has since been updated with further publications.

The architecture is novel, but it is made out of tried-and-true security components. In reality, the team behind the original SDP specification drew on their expertise in the US intelligence community safeguarding sensitive (*high side*) networks.

SDP is intended to address a variety of issues in corporate security, and it has many of the same aims as BeyondCorp and the Zero Trust principles that we've already discussed: *Before gaining network access to protected servers, SDPs need endpoints to authenticate and be permitted*. Then, in real time, encrypted connections are established between the requesting computers and the application infrastructure. Identity-driven network access control, network microsegmentation, and secure remote access are just a few of the applications for SDP.

SDP is an architecture with a variety of deployment models (as well as commercial implementations). These deployment models are quite similar to the Zero Trust models and principles discussed in Chapter 3: *Architectures With Zero Trust*. The *figure 4.3* shows the high-level SDP conceptual model, which depicts the Client-to-Gateway SDP deployment paradigm, which is the most important for our purposes:

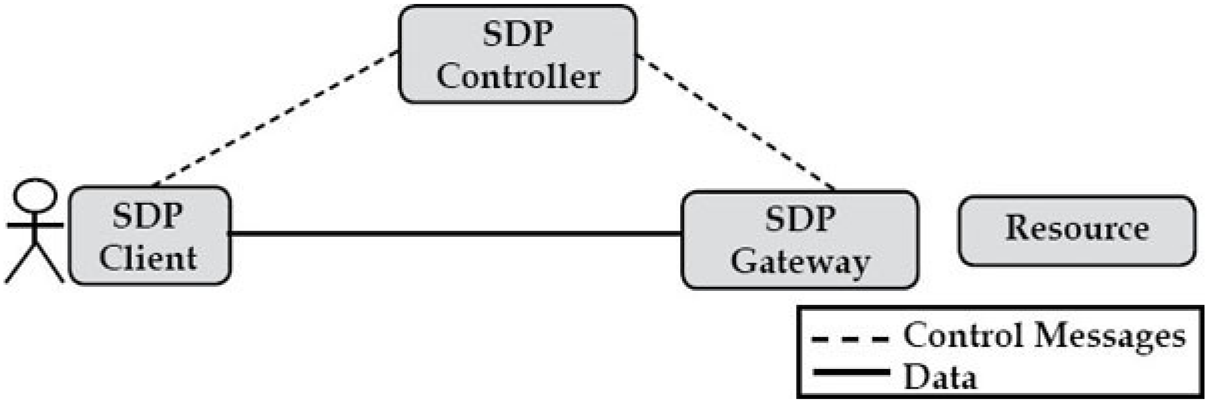


Figure 4.3: *Software-Defined Perimeter Architecture*

There are a few things to keep in mind. To begin, SDP, like Zero Trust, uses separate Control and Data Channels. SDP Gateways are Policy Enforcement Points, while the SDP Controller is a Zero Trust Policy Decision Point. You'll see right away that this SDP model is nearly identical to the enclave based Zero Trust approach discussed in *Chapter 3: Architectures With Zero Trust*. This isn't a coincidence; the NIST Zero Trust team drew inspiration from SDP while developing their architectural document.

SDP requires two security components, mutual TLS communications and single-packet authorization, which we feel should be included in any Zero Trust implementation.

Mutual TLS communications

Mutual TLS Communications (**mTLS**) is a method in which both the client (connection initiator) and the server (connection acceptor) check each other's certificates. This is a big improvement over normal TLS, which only allows the client to check the server's certificate but not the other way around (such as when a browser connects to a web server).

mTLS improves the system's security by effectively removing the chance of a Man-In-The-Middle attack and allowing safe communications even across the most untrustworthy of networks. Of course, it relies on the communication parties establishing a mutual root of trust—a Certificate Authority that both components trust to issue the certificates.

In Zero Trust implementations, mutual authentication, such as mTLS, must be included as a core feature for safe communications.

Single-packet authorization

TCP/IP is a fundamentally open network protocol that was created to make connecting and communicating between dispersed computer nodes simple and dependable. It has served us well in terms of enabling our hyper-connected society, but it does not incorporate security as part of its fundamental capabilities for a variety of reasons. Surprisingly, much of the debate and discussion regarding network security revolves around encryption rather than another flaw: the *connect before authenticate* approach.

As long as the listening device has an open port, any device that can exchange IP network packets with any other device can establish a TCP connection. This is accomplished using TCP's well-known three-way handshake. The most essential thing to remember from a security standpoint is that this connection is established entirely at the network layer, with no identification, authentication, or authorization involved. The beauty of this concept is that it allows anyone with a browser to connect to any public web server on the world and receive a web page without having to register or ask permission first. This is a great concept for a public web server, but it's a bad idea for a private application and a terrible idea for an entrance point with broad access to corporate networks. Despite this, business VPNs operate in this manner, with open ports encouraging unauthorized users to join and exploit vulnerabilities. Unfortunately, this isn't just a theoretical flaw; attackers have exploited it time and time again and have successfully entered business networks because to TCP's open nature.

SDP addresses this flaw by employing a one-time password algorithm based on a shared key in a process known as **Single-Packet Authorization** (**SPA**). In essence, the systems employ an algorithm to produce an OTP and then encode the current password in the first network packet delivered from the client to the server. The SDP standard suggests utilizing the SPA packet after establishing a TCP connection, while the open-source implementation from the SPA designers utilizes a UDP packet before establishing a TCP connection. Both approaches may be used in commercial SDP implementations.

In either scenario, especially with UDP-based SPA, the consequence is dramatic: the servers in issue become invisible to unauthorized clients. Clients that do not submit a valid HOTP will be unable to create a TCP connection and may not even get any indication that a server is listening on the port (depending on the implementation). Authorized clients with the shared key can create a valid HOTP, and the server will allow a TCP connection to be established (followed by a mTLS connection, of course). SPA has another advantage: it requires relatively little processing power for servers to analyses and reject illegal clients. When compared to analyzing authentication after establishing a TCP and TLS connection, evaluating a 64-bit HOTP in a UDP packet consumes orders of magnitude less server resources. SPA-protected servers are thus more resistant to DDoS assaults.

Finally, remember that while SPA is a fantastic initial line of protection, it is only the first layer. Following SPA, which verifies that the client has the shared secret, the SDP system still requires the formation of a mutual TLS connection with certificate validation and identity verification before allowing access to any protected resource.

SDP is a solid architecture that complements Zero Trust. That is, a system based on the SDP architecture may achieve Zero Trust principles. Despite the fact that SDP (as a specification) has a restricted scope, commercially available SDP implementations fill in the gaps and provide an enterprise-ready platform.

Next, we'll look at how one company employed SDP to get to Zero Trust.

SDP case study

In this case study, we'll look at how a US-based multinational company used SDP to start their Zero Trust journey. This firm, which employs over 14,000 workers globally and has been in operation since the 1970s, provides consumer-facing services. Their CISO initiated a deliberate Zero Trust campaign after becoming frustrated with their typical security infrastructure and being inspired by BeyondCorp. His objective was to improve the security of sensitive customer data, cut expenses, and enable the company to use new digital channels for media and customer service. Their infrastructure included two core data centers (one each in the United States and Europe), four US branch offices, eight foreign regional branch offices, and over 700 retail outlets throughout the world.

The firm employed about 2,000 people at their headquarters, another 2,000 people throughout all 12 regional branch offices, and over 10,000 part-time workers at retail stores. Several hundred internally created apps had been transferred from on-premises and were now running in production on the cloud, demonstrating the organization's commitment to IaaS. Their early IT architecture had a lot of flaws, which they planned to remedy with the deliberate implementation of Zero Trust. It's crucial to note, however, that they handled this endeavor in stages, gaining near-instant gratification from their efforts.

In fact, one of this company's assessment criteria was that the security platform they picked should be able to swiftly integrate into their existing infrastructure and enable their transition to Zero Trust over time. For instance, the company was in the process of transitioning from an on-premises Active Directory to a cloud-based SAML identity provider, and its SDP platform required to handle both providers at the same time.

Moving everyone "off net" and using a distributed network of SDP Gateways (PEPs) across their diverse corporate architecture was a crucial component of the Zero Trust effort and vision. The security team looked at a variety of Zero Trust vendors and solutions before settling on an enterprise-class SDP implementation based on the enclave paradigm.

Their first goal was to replace their antiquated and difficult VPN, which was causing connectivity issues and triggering complaints from two groups of users. The first group consisted of about 750 typical business customers who need remote access to resources on the company's office network and data center. The second group consisted of around 250 developers who need SSH, RDP, and database access to development, test, and production resources in an IaaS cloud environment.

Despite the fact that this initial deployment used simple, open policies, it brought immediate benefits by enhancing user experience, increasing connectivity speeds, and giving the security and network teams confidence and experience with their Zero Trust platform. It also allowed developers to access numerous IaaS accounts and locations at the same time while ensuring security. After this step was completed successfully, the security team began limiting access for corporate users on the network by employing group membership from their cloud-based identity supplier. General employee, IT, finance, network admin, and database admin are just a handful of the core positions they designed. Standard services (for example, DNS, print, and file sharing) were made available to all employees, while the other groups were given access to resources unique to their roles.

Then they started disconnecting their 2,000 regional branch office employees from the business network, putting all of their access under the management of Zero Trust regulations. All of the network and security software, hardware, and cabling that had been installed in these 12 branch offices had been removed, and commodity business broadband Internet and Wi-Fi were installed in their stead. Because the vast majority of their production systems were housed in a single data center in the Northeast United States, and corporate users required a secure tunnel to that data center to access business applications, they were able to use existing security software in the data center to perform IDS and SWG functions for users' Internet-bound traffic.

This has the added benefit of saving them over $500,000 per year in infrastructure and communications expenditures. They installed a local SDP Gateway (Zero Trust PEP) in each of their branch offices, allowing users to access local file shares that were regulated by policy. Users had a direct, secure link to the local PEP for access to these file shares thanks to their SDP system design. The in-office user traffic for the file sharing was able to stay entirely on the local network thanks to this architecture. They installed a local SDP Gateway (Zero Trust PEP) in each of their branch offices, allowing users to access local file shares that were regulated by policy.

Users had a direct, secure link to the local PEP for access to these file shares thanks to their SDP system design. The in-office user traffic for the file sharing was able to stay entirely on the local network thanks to this architecture.

The commencement of the COVID-19 pandemic in early 2020 had a tremendous impact on this organization, as it did on all others. All of its 700+ retail outlets throughout the world, which employ over 10,000 part-time workers, were forced to close temporarily. Prior to COVID, these retail workers were linked to centralized application servers via an in-store local wireless network, which was connected to the main corporate data center via a site-to-site VPN from each shop.

The security and network team swiftly shifted gears and installed the SDP client on all of these employees' devices, which included both corporate-managed and BYOD devices. These part-time users were able to work from home right away, assisting the organization as it transitioned to offering services digitally. One unique result for this business was that they were able to start decommissioning the 700+ site-to-site VPNs because all user access was now done over the secure SDP tunnels. As a side advantage, they should be able to save even more money as a result of this.

The next stage in this company's Zero Trust journey is to start installing the SDP client on their Linux servers, employing the microsegmentation deployment paradigm for greater access control across all of their server environments.

Overall, implementing Zero Trust through a SDP architecture has provided this business with clear and compelling security and cost benefits. Because all users are "off net" and the access points into their workplace network are hidden from unauthorized users, their corporate environment is substantially more secure.

The epidemic had essentially little effect on their full-time corporate customers, because nearly all of them were utilizing the Zero Trust solution at the time and were already constantly "remote" from a network standpoint.

Zero Trust and your enterprise

Despite the fact that the first two case studies include internally created systems, we want to emphasize that most businesses, particularly in today's Zero Trust security industry, embrace the strategy followed by the SDP example study. That is, unlike Google and PagerDuty, they license and use commercial software rather than developing it from scratch. Google is obviously in a league of its own as a smart, highly successful, and technologically advanced company, whereas PagerDuty's main business and skill sets focus around managing a complicated, dynamic network. Perhaps most crucially, both of these institutions began their Zero Trust journeys before commercial Zero Trust solutions were generally accessible.

The world we live in now is really different. Every day, both of us writers work with small, medium, and big businesses on their Zero Trust strategy, and they nearly always choose for commercially available or open-source security solutions as the foundation of their platform rather to building their own. Enterprises may and should consider a mix of platform, best-of-breed, on-premises, cloud-based, or hybrid approaches, since there are many viable alternatives available today.

Also, this book is not the best vehicle for analyzing or criticizing vendor or open-source offerings—these products and platforms are always evolving as vendors release new goods and advances or acquire complementary technologies. However, this book is the ideal vehicle for giving you a firm foundation in Zero Trust concepts, as well as an awareness of how it may be implemented in your environment and a set of needs to design, shape, and solidify. Finally, the requirements outlined in this book will help you to make the best decisions possible for your company.

Conclusion

Furthermore, this book is not the greatest vehicle for analyzing or criticizing vendor or open-source offerings—these products and platforms are always developing as vendors release new items and improvements, as well as acquiring complementary technologies. This book, on the other hand, is an excellent vehicle for providing you with a solid foundation in Zero Trust principles, as well as an understanding of how it may be applied in your environment and a set of needs to create, shape, and solidify. Finally, the standards described in this book will assist you in making the finest business decisions possible.

Chapter 5

Identity and Access Management (IAM)

Identity and Access Management (**IAM**) is a wide topic of information security that encompasses both the technical and business process components of access management by granting the correct access to the right person at the right time. Identity—and a relatively well-run identity management program—is, in many respects, the key to a Zero Trust program's success. Because Zero Trust is based on an identity-centric approach to security, knowing and managing identity is a critical component of any Zero Trust programme. Yet, before commencing on their Zero Trust journey, enterprises should not and cannot subject themselves to an impossible standard or demand perfection from their identity teams and systems.

Identity management systems should be utilized as the *keystone* system for various technological integrations and business processes, serving as the authoritative source of information about identities (both person and non-person entities). This is difficult since today's businesses are complicated, and there may not be a single, centralized identification system in place. That's OK, and it shouldn't prevent you from using Zero Trust. In fact, because it is an overlay system, Zero Trust may assist bridge the gap between numerous identification systems. We'll go over this in more detail later in this chapter, in the section on IAM and Zero Trust. But first, we'll go through the main components of IAM, which are the foundation for all of the book's discussions regarding Zero Trust.

Review of IAM

While each identity management system is unique, owing to the unique mix of each company and its selected collection of technologies, all identity management systems share the characteristics represented in *figure 5.1*. Let's take a look at each of the categories to get a sense of the range of IAM programmes:

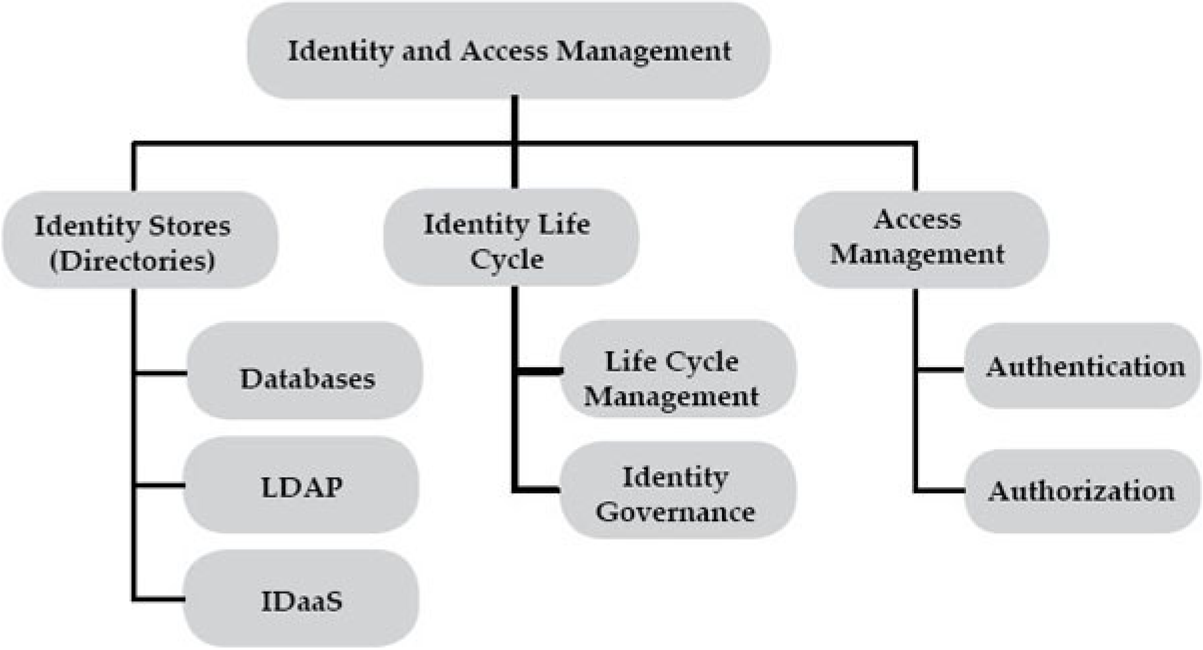


***Figure 5.1:** Identity Management System Scope*

Identity stores (directories)

The identity store, often known as a **directory**, is the heart of any identity management system (more formally, a directory service). This is conceptually where authoritative information about entities is stored attributes that characterize the thing and offer relevant data about it for human and machine consumers of this information.

Enterprise IT use of PC-based local area networks prompted the formalization of directories in the late 1980s. These directories served as a searchable and authoritative list of information about users and were used to authenticate people for network access. User credentials were provided, making directories a consolidated, centrally maintained source for user authentication.

We believe it is fair to say that the whole contemporary IAM ecosystem has developed from this fundamental directory function. It, like many other fields, has grown increasingly standards based as time has passed. The X.500 specification was used to store entity information in directories, with initial connection provided by the **Directory Access Protocol** (**DAP**).

This was not based on TCP/IP networking and was extremely difficult to use for clients, resulting in low acceptance. As a result, a "*lighter*" version of DAP has been developed: We'll talk about **Lightweight Directory Access Protocol** (**LDAP**) shortly.

Over the last few decades, directories and the identity management systems that surround them have definitely evolved in capabilities and scale. Many various and sophisticated scenarios are supported by today's directories, including metadirectories and federated directories, which connect numerous separate directories (albeit in different ways).

Following that, we'll go through the three most common forms of directories in use today.

Databases

Databases can theoretically serve as a centralized identity storage that can be accessed from anywhere on the network. For a variety of reasons, most modern businesses have shied away from utilizing raw databases as directories. Even if it's read-only, giving remote programmes database access to user information (particularly credentials) is bad design.

Even standards-based directories, in the end, are dependent on an underlying database. Raw database access, on the other hand, differs significantly from defined protocols and APIs for dealing with directories. Customized identity stores should be avoided, and if they do exist, they should be phased out as part of a Zero Trust programme.

Lightweight Directory Access Protocol (LDAP)

LDAP is a protocol definition that defines a series of messages (essentially an API) for communicating with directory services across a network. The

Chapter 5: Identity and Access Management (IAM), described LDAP as a well-established standard in a series of RFCs.

In the sense that numerous directory providers (both open source and commercial) implement the protocol, and components from various manufacturers may effectively interoperate, LDAP v3 was a fairly successful standard when it was first introduced in 1997.

LDAP provides a simple API for working with the directory's set of entities, and it's also widely used to authenticate users' credentials (passwords).

Today, LDAP is extensively used and maintained, with support from a wide range of identity, security, application, and infrastructure providers. Microsoft's Active Directory, for example, has an LDAP API and is likely the most extensively used directory in the business.

While we expect LDAP-enabled directories and applications to continue to function for a long time, we predict that newer, standards-based authentication and authorization protocols will eventually displace LDAP. Modern protocols enable indirect token-based procedures that are more suited to today's dispersed environment than LDAP, which needs direct API calls to the directory.

However, one or more LDAP-based directories are likely to remain in existence in your organization, and your Zero Trust platform must be able to interface with these services without asking them to change. As long as LDAP fits your functional requirements, there's nothing wrong with continuing to utilize it.

Identity-as-a-service

Identity management suppliers, of course, have benefited from the change to cloud-based services, resulting in a profitable and fast increasing market segment known as **Identityas-a-Service** (**IDaaS**). These companies offer cloud-based directories that eliminate the need for on-premises directory servers, provide a contemporary web-based user interface, and include capabilities like **Single Sign On** (**SSO**) and, increasingly, passwordless authentication.

These services usually provide both modern APIs like SAML as well as older APIs like LDAP and RADIUS. As cloud-based security services gain acceptability and maturity, these services are well positioned for sustained development. It's worth noting that many of these providers offer on-premises software (agents) to execute certain tasks, such as federation or data replication, as well as interaction with older directories and authentication mechanisms.

Every company will eventually have one (or more) identity stores. Organizations require a mechanism to standardize and normalize data across numerous heterogeneous identity stores, and Zero Trust solutions must interface with them. This is especially true when it comes to safeguarding access for third-party identity storage. When we talk about authentication in the access management part of this chapter, we'll go over this in more detail. Prior to that, we'd want to discuss the identity lifecycle, which is another crucial aspect of IAM.

Identity lifecycle

Whether expressly and officially stated or not, every identity has a lifetime.

Identity is created, it exists for a length of time, it may play diverse roles over time, and it is finally destroyed. To manage and govern identity lifecycles, businesses need technological tools and business/IT procedures. Lifecycle management and Identity Governance, two elements of IAM, are inadvertently part of a Zero Trust project.

Lifecycle management

The identity lifecycle is sometimes referred to as "*Joiners, Movers, and Leavers*" when discussing human users.

In most cases, a human being (user) passes through the lifespan shown in *figure 5.2*. This includes granting "*birthright*" credentials, which are automatically granted when a user (Joiner) is joined to the corporate directory for the first time. Users usually have access privileges that go beyond these, which should be allocated depending on their position. Organizations must exercise caution when giving users unnecessarily wide permissions, especially when utilizing an existing user's privileges as a template for new users (the "*make Sally seem like Jimmy*" problem).

As a result, people will have more access than they need, possibly posing security and compliance risks. As we shall see momentarily, a well-run identity management programme will prevent this problem through roles and identity governance:

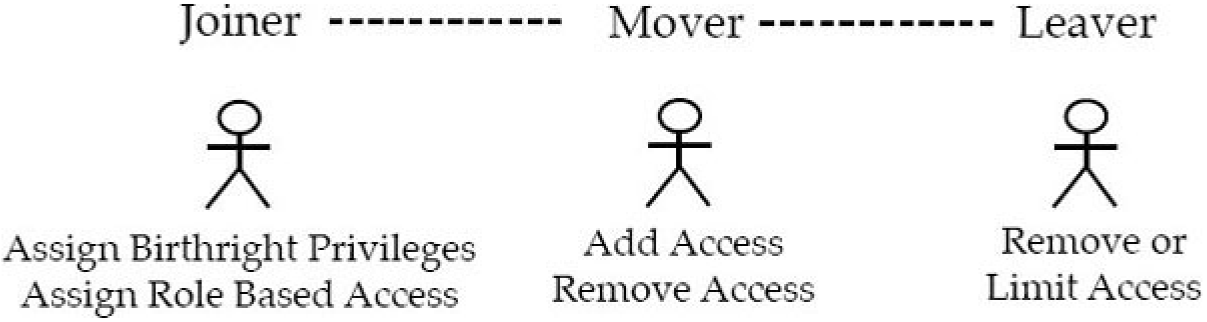

Joiner ------------ Mover ------------ Leaver
Assign Birthright Privileges
Assign Role Based Access
Add Access
Remove Access
Remove or
Limit Access

Figure 5.2: *The User Identity Lifecycle*

As a user (Mover) progresses within an organization, either laterally or hierarchically, their access changes as they get new privileges as part of their new job. Adding access is simple—there are apparent triggers, since the person will need (and demand) access in order to do their new task. Removing access is more delicate, especially since most users require both new and old access during the transition phase. Because this phase might last weeks or months, corporate processes to track and manage it are required (usually as part of an identity governance programme).

Leavers are users who are at the end of their lifespan. This phase might begin for a variety of reasons, including planned voluntary departure (resigning or changing jobs) or involuntary leave (being fired) (immediate termination). In many circumstances, individuals will remain in an identity management system for an extended length of time, for example, so that managers may access the email of a departing employee. In certain circumstances, departed users may remain in systems for long periods of time (for example, to access personal payroll, insurance, or tax information), or even indefinitely (for example, a student transitioning to an alumni of an educational institution).

Based on these lifecycle events, an identity management system must manage, and preferably automate, the assignment, provisioning, and deprovisioning of user access. HR and payroll onboarding and offboarding are handled well by the majority of companies.

The linked IT procedures, on the other hand, are frequently immature and ineffective. People are rarely compensated when they leave a company, for example, according to anecdotal evidence. However, it's extremely usual for users to keep access to IT systems after they leave (particularly for SaaS apps).

Non-human accounts (service accounts) need a somewhat different level of vigilance, as these systems are often separated from any external or HR-driven events, such as hiring or firing.

- Joiner Mover Leaver accounts are just accounts that are used for service.
- Assign Birthright Privileges and Add/Remove Access
- Assign Access Based on Roles
- Access should be removed or restricted.

Graph 5.2: The Identity Lifecycle of a User are designed for usage by servers or infrastructure code rather than by humans.

Accounts, as well as other access control methods such as API keys or certificates, are examples of non-human access mechanisms.

Like regular user accounts, these accounts have privileges and roles associated with them, which must be actively managed. Also, like user accounts, these accounts must have only the minimal set of privileges associated with them. This is often more of a challenge for service accounts, since they are often performing system-level activities, and there may not be a robust model in place to limit their privileges in the target system. Too often, these accounts are granted full admin rights, either by necessity or by the need to solve a problem quickly (*Don't worry... we'll fix it later*). And these account credentials are often shared, stored in clear text, and typically not rotated. The clear conclusion is that service accounts must be included in identity governance processes, just as user accounts are, and that Zero Trust systems should be used to enforce access policies for these accounts. Note that **Privileged Access Management (PAM)** solutions often do have service account vaults and services that can help solve some of these problems. We'll be covering PAM in a later chapter.

Identity governance

Identity governance refers to the policies (or, if you prefer, rules) that define "*who should gain access to what*" as part of the previously stated identity lifecycle. It is another aspect of the normal scope of IAM programmes. Compliance and security requirements are frequently used to drive governance, with compliance being the primary motivator. In many situations, especially for publicly listed organizations, these compliance obligations are focused on financial applications and controls.

Identity governance products have been developed by vendors in this industry to help satisfy these compliance requirements, and companies often install these solutions as part of their identity governance activities. Of course, not all businesses have formal identity governance processes in place; smaller, less regulated businesses may not require them.

However, as part of the identity lifecycle procedures, all businesses make judgments about *who should have access to what*, whether tacitly or explicitly. These decisions will eventually manifest themselves as modifications to underlying software systems, such as changes to user characteristics or group memberships in directories, or the creation, deletion, or permission of user accounts within programmes.

These access control choices can be made automatically by a provisioning system or manually by IT and business procedures. Identity governance principles must be linked with Zero Trust policies in any circumstance. When we cover authorization in the next chapter, we'll go through this issue and the interaction between these levels.

Access management

The two fundamental components of access management are authentication and authorization. Authentication is the technique by which entities establish that they are who they claim to be, and authorization is the model for defining and expressing the set of activities that a particular entity is authorized to undertake. Let's take a look at each one separately.

Authentication

We give a quick overview of common authentication methods, processes, standards, and trends in this area. We're doing this to see if they're relevant in Zero Trust deployments. Let's start with some fundamental definitions, which we've supplied for your convenience:

- **Password/username**: For decades, simple authentication has been in use. This is the concept of verifying what you already know.
- **MFA (Multi-Factor Authentication)**: The use of several authentication factors as part of a single authentication procedure. This usually involves using a physical token, a smartphone app, or a biometric technique to verify what you have or who you are.
- **Step-up authentication**: When a user is prompted for an extra form of authentication following an event or trigger, this is known as step-up authentication.
  This might be triggered by a user who has previously been authenticated attempting to access a high-value resource. Because the user has previously been validated, this frequently employs some type of MFA.
- **Passwordless authentication**: Using elements other than passwords for initial authentication is a generic principle. This move is welcomed and encouraged since it eliminates the well-known hazards of weak passwords, password theft, and password repetition.

These solutions frequently employ the processes described under MFA. Let's have a look at some of the existing authentication methods and procedures.

LDAP

We recently discussed LDAP, which is an API that can be used to communicate with directories as well as authenticate users. The LDAP API contains native support for login and password-based authentication, which is the most prevalent method of authentication. Other authentication methods can be added to the LDAP API using a challenge-response system, which is included in the API. These are often used; however, their use is depending on the implementation (non-standardized).

RADIUS

Remote Authentication Dial-In User Service (**RADIUS**) is another older authentication system whose antiquity is evident in its name. It was originally designed to offer **Authentication, Authorization, and Accounting** (**AAA**), which are essentially the forerunners of today's Identity Management. While the acronym AAA is no longer widely used, similar principles are still present in today's mainstream Identity and Security programmes.

RADIUS is still commonly used in practice today, despite its antiquity.

Many vendors accept it since it is part of multiple official IETF standards (RFCs) and provides reasonable compatibility with inter-vendor components. RADIUS, like LDAP, has a basic concept in which a RADIUS client (commonly referred to as a "*network access server*") communicates directly with a RADIUS server to provide authentication on behalf of a principal (usually a user). RADIUS will only respond with an accept or refuse (perhaps after a second challenge, enabling MFA). Although we haven't seen this in widespread use, RADIUS may be used to offer identity context via protocol extension.

RADIUS, on the other hand, does enable standards-based authentication procedures in addition to username and password, and this has undoubtedly prolonged its longevity.

Many current identity providers, in fact, provide RADIUS APIs or Gateways, which allow older applications or infrastructure to be linked into newer platforms. Older systems can leverage the newer MFA or passwordless authentication mechanisms enabled by contemporary identity platforms, which can be triggered via RADIUS, using this strategy.

Security Assertion Markup Language (SAML)

The **Security Assertion Markup Language** (**SAML**) was born out of a need and need in the industry for a trustworthy, trusted, and interoperable mechanism to provide SSO for users into online applications, particularly for web apps from various providers.

In more technical terms, SAML specifies an XML representation and an HTTP-based protocol via which web applications (referred to as "*service providers*" in SAML) can receive user authentication and attribute data from a separate **Identity Provider** (**IdP**). That is, in addition to authenticating the user, the SAML response data (also known as **assertions**) might include extra information about the user that was requested by the web app and provided by the identity provider.

SAML has been a huge success as a standard, with widespread adoption across identity providers, SaaS and private web businesses, and a thriving industry for SSO from Identity-as-a-Service providers. It's a fantastic example of a network effect, where the value of supporting the standard rises with each additional supporting participant, as it's a simple and stable standard that enables an easily setup trust connection between web apps and IdPs. With so many open-source toolkits and plug-ins available, there's no reason why web apps shouldn't support SAML. Similarly, Zero Trust solutions must accept SAML as an authentication mechanism and welcome the widespread deployment of SAML-enabled identity providers.

With extensive acceptance across identity providers, SaaS and private web enterprises, and a growing sector for SSO from Identity-as-a-Service providers, SAML has been a major success as a standard. Because it's a simple and stable standard that provides an easily created trust link between web apps and IdPs, it's a perfect example of a network effect, where the value of supporting the standard improves with each new supporting participant. There's no reason why web apps shouldn't support SAML, especially with so many open-source toolkits and plug-ins available. In the same way, Zero Trust solutions must support SAML as an authentication technique and welcome the broad use of SAML-enabled identity providers.

OAuth2

OAuth2, an IETF standard, is intended to allow developers to construct authorization protocols that allow a third-party application to access a restricted range of functions or resources within a web application on behalf of a user. For example, a user might allow access to their private photographs on a photo-sharing site to a photo-printing business without having to reveal their account and password. In more formal terms, OAuth2 allows a client to get a security token from a trusted token service (usually an IdP) and send it to a relying party for usage. It's important to note that this is mostly dependent on the user's consent. OAuth2 is a mechanism for authorization rather than authentication. The authentication protocol OpenID Connect, which we'll explore next, is built on top of OAuth2.

OpenID Connect (OIDC)

OIDC is based on OAuth2, and its token is a **JSON Web Token** (**JWT**). It's intended to provide authentication to OAuth authorization, and it's most commonly used by web apps to give authentication and authorization using the underlying OAuth framework, which is built on an interoperable REST standard. For usage within the target application, an OIDC token contains trustworthy assertions about the user (relying party).

Certificate-based authentication

Certificates (and their accompanying technologies) are frequently used in business identity management to confirm user and device identities—having a valid certificate may positively identify an entity. In practice, this means that each user's device has a valid and current certificate installed on it, that this certificate was issued by the organization's own Certificate Authority (as part of its **Public Key Infrastructure** (**PKI**)), and that the user can log into the desktop or mobile OS in such a way that the certificate—secured in the OS's local key management system—is accessible to the user's account. Non-user devices, such as servers or IoT devices, each have their own procedure for installing certificates and using them for identification and authentication by the owning company. Finally, some physical identity cards have enterprise-issued certificates that may be accessed with a PIN and used as a kind of multifactor authentication. The **Common Access Card** (**CAC**) and **Personal Identity Verification** (**PIV**)

Card are common examples of this in the US Government and Defense sectors.

FIDO2

Through the FIDO **Universal Authentication Framework** (**UAF**) and two Client-to-Authenticator-Protocols versions, FIDO2 is delivering the "*passwordless*" experience to the end-user community (CTAP1 and CTAP2). FIDO2 enables browsers, mobile devices, and hardware fobs as authentication methods thanks to these PKI-based protocols.

Mobile and biometrics

Modern authentication techniques increasingly incorporate user-friendly and/or mobile device-based technology for user authentication, however they are not authentication standards. End customers are familiar with technologies like fingerprint recognition or face recognition on their mobile devices, and mobile apps for generating **One Time Passwords** (**OTPs**) have mostly supplanted their hardware token-based counterparts.

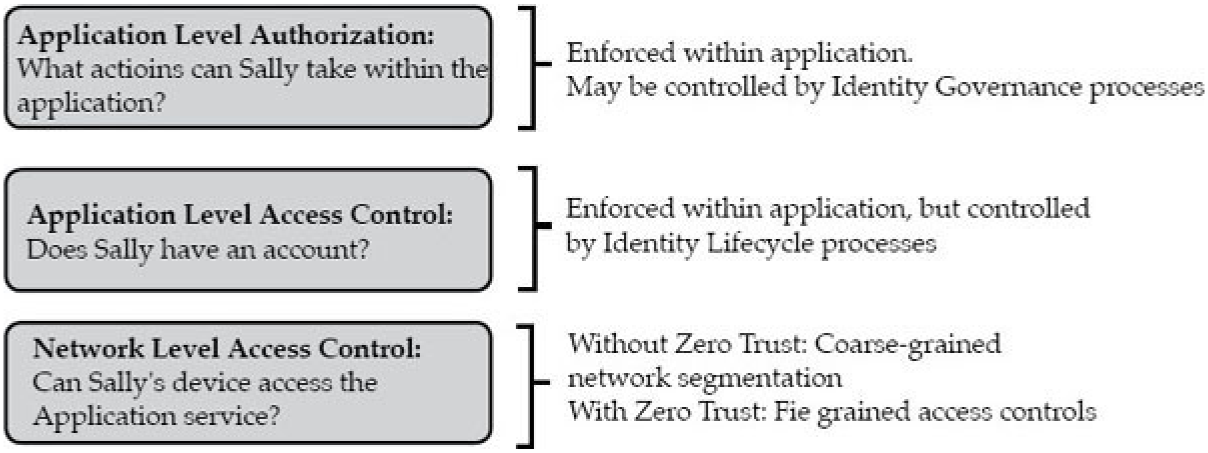
Mobile devices provide a dependable and user-friendly manner of implementing a second factor, which is a significant aspect of Zero Trust. Indeed, given the remarkable ease-of-use that consumer-facing apps (such as contactless payment) today deliver, end customers want the same degree of simplicity from their workplace IT systems.

Unfortunately, doing this isn't always easy or straightforward, since business IT is attempting to solve a more complicated challenge than the authentication and authorization of a one-time credit card transaction. For minutes or hours, enterprise security systems generally authenticate individuals and authorize access to business or technical applications. Enterprise IT is sometimes impeded by a more complicated network infrastructure with several limitations. Having said that, these standards and technologies perform effectively, and we are seeing a shift in the acceptability and prioritization of utilizing newer authentication methods rather to passwords. With their intrinsically dynamic and comprehensive scope, as well as their deep integration capabilities, Zero Trust security architectures are assisting in this.

Authorization

After all, access management is ultimately responsible for mapping from some policy model (technical or business policy) to enforcement points, and authorization is the final aim. Identity management systems, when implemented correctly, provide authoritative characteristics—roles and attributes—that are connected with entities, as well as governance policies and processes to verify that these are valid. Of course, these characteristics are only useful if there exist runtime IT components capable of appropriately enforcing these regulations. For example, simply because Sally is a member of the "*astronauts*" directory group does not guarantee she is an Astronaut.

Sally's membership in the directory group "*ABC123*" is also meaningless. What matters in both circumstances is how this information is interpreted by the rest of the IT, application, and security systems, and how it impacts Sally's set of accounts and access. Authorization happens at various levels in practice, as shown in *figure 5.3*:



Application Level Authorization:
What actioins can Sally take within the application?
Enforced within application.
May be controlled by Identity Governance processes
Application Level Access Control:
Does Sally have an account?
Enforced within application, but controlled by Identity Lifecycle processes
Network Level Access Control:
Can Sally's device access the Application service?
Without Zero Trust: Coarse-grained network segmentation
With Zero Trust: Fie grained access controls

Figure 5.3: *Access Control Levels*

The application-level authorization model, which controls what activities our user (in this example, Sally) may conduct within the application, is at the top. This is usually implemented directly inside the application9, depending on Sally's account's roles or permissions. While it's probably easiest to think of the "*application*" as a business application, such as a financial management system, this model can also be applied to a more technical application, such as a source code repository or database system, or even a completely different type of service, such as an online store.

Login to a server via SSH. In any event, more mature companies will have a set of identity governance procedures and technologies in place to assess and enforce application-level authorization. In less established businesses, this is usually done on an ad hoc basis, depending on basic or preset application roles.

The application account level is the intermediate layer, which generally controls application access by the presence or absence of a certain user’s application account. In order to access the programme, this layer needs the user to submit valid credentials. That is, authentication is used to control access. Many access control solutions, such as SSO and PAM solutions, operate at this tier.

These first two levels represent identity management’s conventional scope and limits, with a generally obvious demarcation from the underlying layer of network-level access control. Without Zero Trust, security and networking teams could only enforce access control in a static, coarse-grained manner, such as assigning users to an entire Virtual LAN (VLAN) with hundreds or thousands of hosts, remotely connecting entire networks with WANs like MPLS or site-to-site VPNs or granting remote users full network access via user VPNs. With Zero Trust, the network layer may implement fine-grained access rules based on roles and characteristics, which are only available and effective at the application layer in typical security systems.

As we wrap up our IAM in Review part and go on to Zero Trust and IAM, we’d want to talk about RBAC and ABAC quickly. These phrases refer to a generic approach to access control that is based on traits linked with an identity, which are often retrieved via an identity management system. (Technically, a role may merely be thought of as a type of attribute, hence ABAC can be thought of as a subset of RBAC.) The capacity of an organization to set access policies, which can specify logical constraints under which an identity is authorized to access a specific resource, is referred to as the “control” aspect of ABAC.

If this sounds familiar, it's because Zero Trust implements attribute-based restrictions.

Indeed, we believe that Zero Trust architectures are the most efficient method to implement attribute-based access control. At the end of the day, ABAC is only a notion that has to be defined in a physical architecture, on a platform that provides organizations with a rich language to describe these access control policies. Any Zero Trust programme should be built on this policy paradigm, with its scope, capabilities, and efficacy. We'll go through this in greater detail later.

Zero Trust and IAM

Now that we've gone through IAM and its components, we'll look at why and how IAM is so important to Zero Trust. Remember that the Zero Trust PDP uses IAM systems to authenticate entities and to provide context (roles and characteristics) for policy choices. As previously said, we are fortunate that our industry has developed a number of standardized authentication APIs and protocols. These have been widely adopted to the point that we can rely on our identity provider and Zero Trust platform to work together.

Along with leveraging current IAM systems for user authentication, Zero Trust platforms must be able to collect identity context from them so that PDPs may make access choices. This emphasizes why Zero Trust solutions must support common protocols (in particular, LDAP and SAML).

These principles apply regardless of which Zero Trust deployment model (or models) is used. Zero Trust solutions must always be coupled with identity systems; this is what makes this approach to security so much more successful than previous methods. Let's compare Zero Trust to old ways for a moment, which we feel will demonstrate not just identity integration and Zero Trust, but also the total benefit of Zero Trust in the enterprise.

Authentication, uuthorization, and Zero Trust integration

The *figure 5.4* shows three instances in which a user is interacting with an application. The user has an account with the programme, must authenticate, and has a set of rights that they may utilize within the application in all instances. If this is a website **content management system (CMS)**, for example, the user may be able to update pages but not submit altered pages into production. What's fascinating, though, are the variations in security and integration across the three situations:

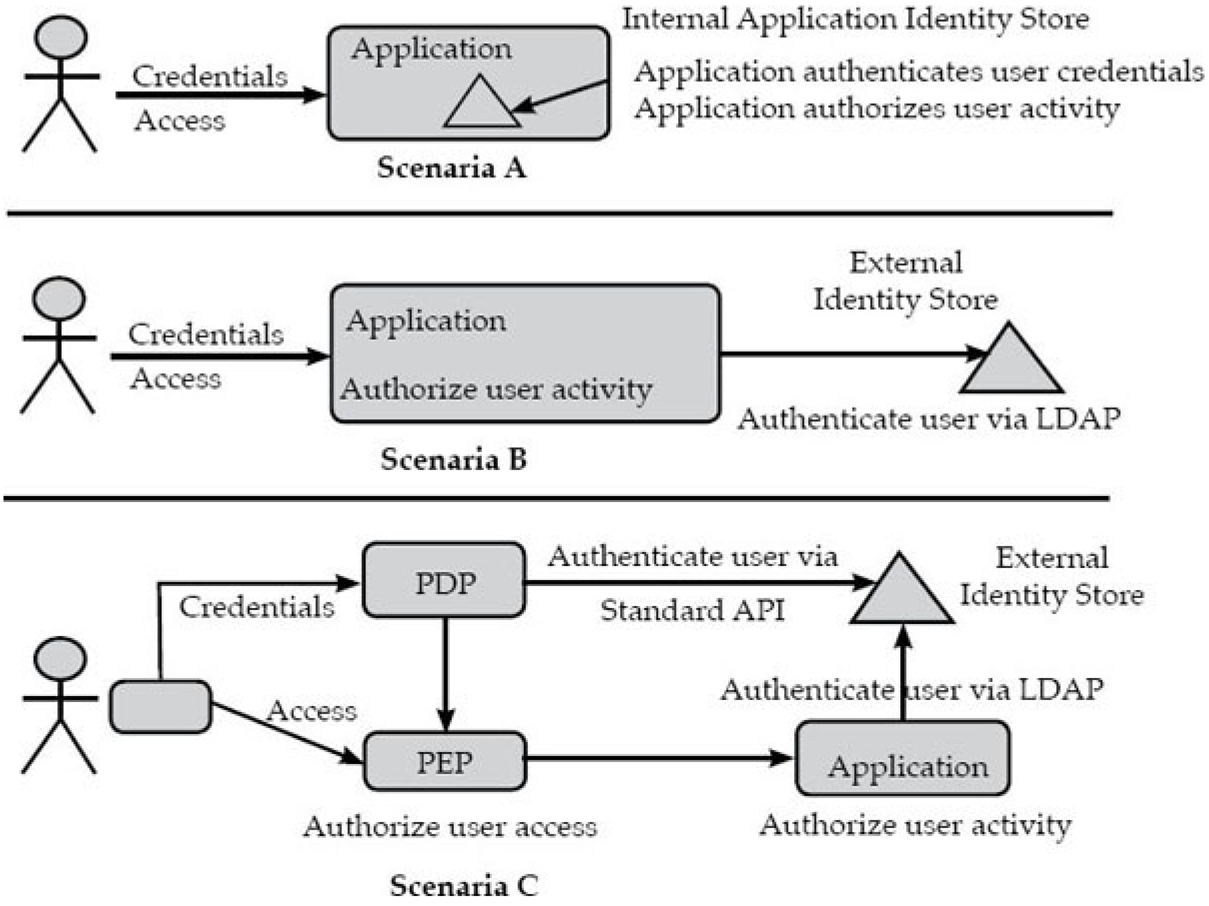


Figure 5.4: *Authentication, Authorization, and Zero Trust*

Scenario A depicts a traditional, stand-alone application with its own internal identity and credential storage. Users log in directly through the app, which also enforces user permissions. While this method works and innumerable apps have been created in this manner, it has a number of drawbacks. It's the definition of a "*silo*" as a stand-alone and self-contained identification system. Not only does bespoke code like this typically have security flaws that might be exploited, but it also runs the danger of being left out of mover or leaver lifecycle events, leaving accounts active but unused ("*orphaned*"). It's also possible that it's not utilizing an encrypted network protocol or that it doesn't support MFA.

Scenario B's application has obviously improved in several aspects. It avoids becoming a silo by relying on an external, LDAP-based identity system, and it will immediately integrate with the organization's centralized identity governance and lifecycle procedures.

MFA may be supported by the LDAP system, enhancing authentication security. This application, like Scenario A, may employ an unencrypted network protocol. This programme (together with other services operating on the same server) is most likely visible to any user on the network, just as it is in Scenario A. As a high-value programme, our online CMS is a tempting target for a hacker—imagine being able to inject malicious code into a company's official website!

***The application** is depicted in Scenario C under a Zero Trust security architecture.*

While the programme still uses LDAP to authenticate users, network access to it is now secured by a PEP. This restricts access to that host throughout the network to only authorized users, making it far more difficult for the adversary to attack. Furthermore, not only may authorized users use the programme remotely, but network traffic between the user's device and the application's PEP is encrypted. Based on policy and user circumstances, the Zero Trust system is likely to enforce MFA when needed.

Users may even be able to be automatically authorized into the application via SSO, depending on the capabilities of the identity and Zero Trust systems.

Increasing the authentication of legacy systems

One of the most intriguing and unique features of a Zero Trust system is how it may enhance the reach and value of authentication methods that are generally limited in scope. Because Zero Trust systems are connected with identity management systems and can enforce regulations at the network layer, they open up new possibilities for authentication. Take, for example, the "*legacy*" application shown in *figure 5.5*:

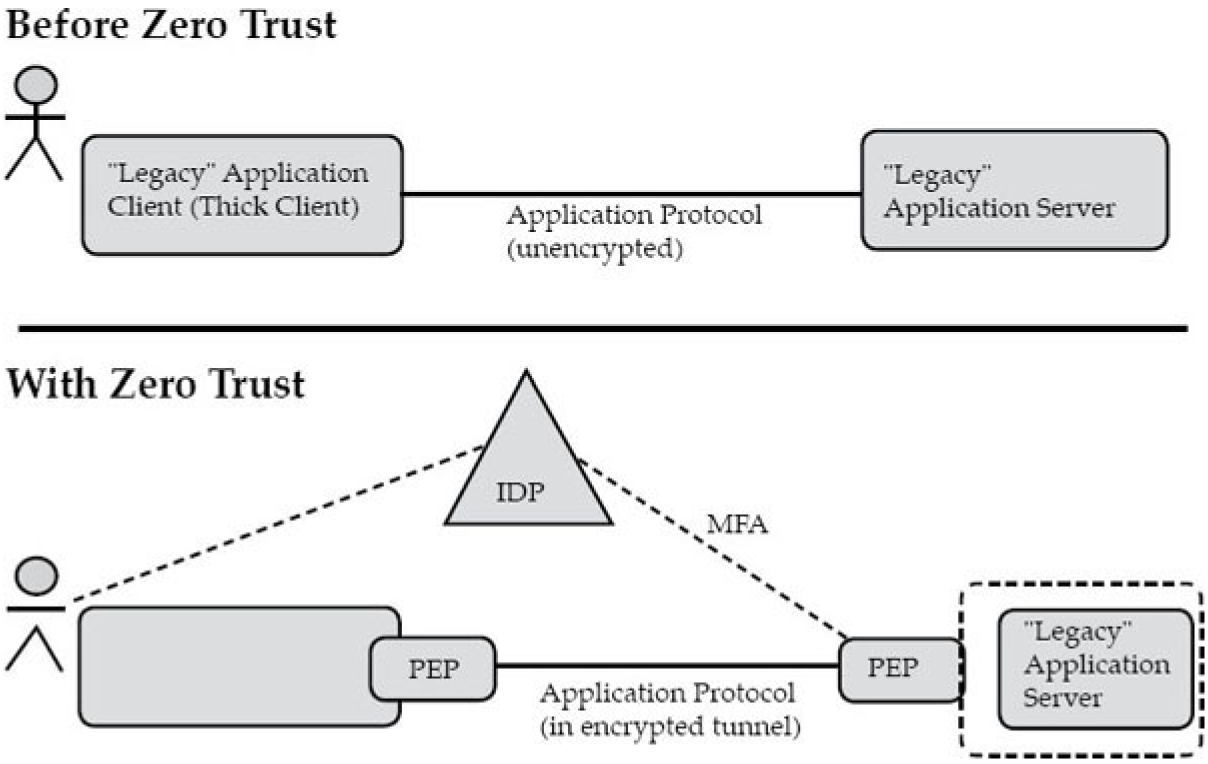

Before Zero Trust
"Legacy" Application Client (Thick Client)
Application Protocol (unencrypted)
"Legacy" Application Server
With Zero Trust
IDP
MFA
PEP
Application Protocol (in encrypted tunnel)
PEP
"Legacy" Application Server

Figure 5.5: *Before and After—Legacy Application Authentication*

Users access this core business application via a thick client in the "before" state, which employs an unencrypted application-specific protocol. Despite the fact that this communication is travelling over a regular TCP/IP network and originates from a conventional enterprise-managed device, it does not use contemporary application protocols (such as HTTPS), making it unavailable to modern tools and security systems that rely on HTTP headers. Despite being "*closed*" to security tools, the programme delivers unencrypted important business data. Furthermore, this business is unable to make this programme web- or SAML-friendly. 10 All of these reasons make meeting their security and compliance requirements of requiring MFA and encrypting network traffic difficult. The firm has implemented a contemporary authentication system in the "*with Zero Trust*" stage. For both initial user logins and enforcing MFA, this system serves as the official identity source. Despite the fact that this historical application isn't connected to the IDP in terms of identity or primary authentication, the Zero Trust PEP can intercept the user's access to the application and notify the IDP to impose MFA before allowing the user's access to proceed. This method has a number of key advantages for businesses: First and foremost, they comply with the MFA and encryption standards.

Second, they utilize the same IDP throughout the organization, simplifying the user experience and lowering operational complexity.

And the Zero Trust approach enables businesses to accomplish all of these objectives without modifying the application server or client.

Even with in-place aspects of an IT infrastructure that cannot be modified, the above example illustrates one method in which Zero Trust principles and advantages can be accomplished. Zero Trust architectures, in particular, are particularly positioned to offer this sort of benefit while minimizing disruptive changes as an overlay onto existing networks. Let's talk about how Zero Trust solutions may assist enterprises develop their IAM initiatives while we're on the subject of transformation.

Zero Trust as catalyst for improving IAM

Zero Trust initiatives provide a fantastic chance for enterprises to upgrade or alter their identification systems gradually or drastically. When addressed correctly, Zero Trust initiatives enable enterprises to streamline and improve their existing identification systems, or transition to more current and effective solutions.

Many bigger businesses, for example, have many incompatible directories for authentication and user characteristics in place. These systems may have developed separately over time, maybe as a consequence of distinct departments with specific project requirements, or as a result of an acquisition.

These might include cloud directories, customer identification systems, or even business partner identities, and are not confined to directories inside a single infrastructure.

Consolidating all of these directories into a *one Directory to Rule Them All* may be a worthwhile endeavor, but it should not be used as a deciding factor before starting a Zero Trust project for two reasons.

First, many of these diverse identification systems have different and sometimes contradictory sets of requirements, making it difficult for a single identity system to meet all of them. These distinctions might take the shape of technical platform or integration needs, support for unique regulatory or compliance criteria (such as data residency), or even something as basic (but crucial) as local language support.

Second, companies should regard the Zero Trust initiative not just as a stimulus for replacing outmoded technology (which we'll go into in depth in later chapters), but also as a vehicle for standardizing different systems.

When implemented correctly, Zero Trust may simplify security and operations by functioning as a homogenizing layer that hides the underlying complexity—almost like a blanket of snow smoothing out a rough terrain.

That isn't to suggest that Zero Trust programmes can compensate for (or miraculously cure) fundamentally faulty identification systems. Most identity teams, on the other hand, are led by brilliant and driven individuals who are juggling a complicated set of tools and a big amount of work.

Without needing wholesale or disruptive changes, Zero Trust can assist simplify and speed identification processes, as well as minimize the overall complexity of the identity programme.

Conclusion

Identity management systems include a wide range of topics, as we've seen in this chapter. They're usually complicated—inherently dynamic and frequently clumsy—because they deal with Joiner, Mover, and Leaver procedures on a daily basis, exceptions and all. This complexity is inherent in reality—identity systems are basically a software and process model of the company, its employees, and their responsibilities. From this perspective, it's not surprising that businesses form specialist teams to manage these systems and that a vendor and consultant ecosystem has grown up around them.

Although the identity lifecycle (which includes identity governance) is ultimately responsible for determining *who should have access to what* (that is, authorization), it is typically reliant on manual or automated IT systems for account provisioning11, which is how application-level access control is enforced.

Zero Trust systems implement access control at the network level, and in order to do so, the Zero Trust system must be able to retrieve identity characteristics for use in its policy model at the time of identity authentication and on a regular basis thereafter. Zero Trust teams must be purposefully connected with identity management teams, which may be done through organizational design. Based on data from identity systems, Zero Trust systems impose access restrictions (some of which will come from the identity team). This will rely on defined and reliable APIs, database schema, versioning, and change management, just like any other inter-system interaction. Yes, this is labor, but it shouldn't be regarded as onerous. IAM procedures must exist inside organizations—people join, move, and depart on a daily basis—and security teams must enable these operations effectively, allowing users to remain productive while maintaining security. Identity is at the heart of Zero Trust, and there is a wide range of mature and new standards that may be used to properly integrate these parts.

Because you'll be using your organization's IAM systems for authentication and identity attributes, learning how they function is a natural aspect of any Zero Trust endeavor. Identity management programmes (technology, people, and procedures) can be useful for your Zero Trust campaign, even if they are still in their infancy—your IAM environment doesn't have to be flawless (but it also mustn't be "*broken*"). In the end, regardless of where their IAM systems started, enterprises can and should adopt Zero Trust.

CHAPTER 6

Network Infrastructure

An organization’s journey to Zero Trust will undoubtedly have an influence on network infrastructure, as well as the hardware and software infrastructures that support it. In fact, a big part of Zero Trust’s power and value is its ability to implement identity and context aware regulations at the network level, linking these two realms that are usually distinct. As a result, a switch to Zero Trust will have an impact on business network architecture, operations, and perhaps network topology. This is something that security and network architects should be aware of, so they can plan for it. Few businesses would start from scratch, thus security architects designing for Zero Trust should work with their IT counterparts to gain a better understanding of their present environment.

Existing infrastructure components must inform and impact an enterprise’s Zero Trust design and needs while imposing as few limits as possible. As a result of implementing Zero Trust, networks and their teams, operations, and procedures will need to adapt. Rather of fighting these developments, we must welcome them. We understand that this is easier said than done, and that cultural changes might be more difficult than technological ones.

Even modifications that seem simple and can be implemented “like for like,” such as replacing a VPN with a Zero Trust remote access solution, might be difficult to implement organizationally or politically. We’ll look at some of the nontechnical features of Zero Trust installations in *Chapter 19: Creating a Successful Zero Trust Environment*, but for now, we’ll concentrate on the impact of Zero Trust on core network infrastructure components such as firewalls, DNS, and Wide Area Networks. We also mention Web Application Firewalls, API Gateways, and Load Balancers/Application Delivery Controllers in passing. Other network features, such as NAC and VPN, have a lot more to say, and we’ll cover them in other chapters. Finally, keep in mind that, with the exception of NAC in *Chapter 7: Network Access Control*, we've mostly avoided discussing network hardware and switching or routing.

Firewalls for the internet

Of course, network firewalls have long been the bedrock of network security architecture, acting as the first *policy enforcement points*. Firewalls will undoubtedly exist in a Zero Trust network, but their function will alter. As shown in *figure 6.1*, we believe one of two scenarios will occur:

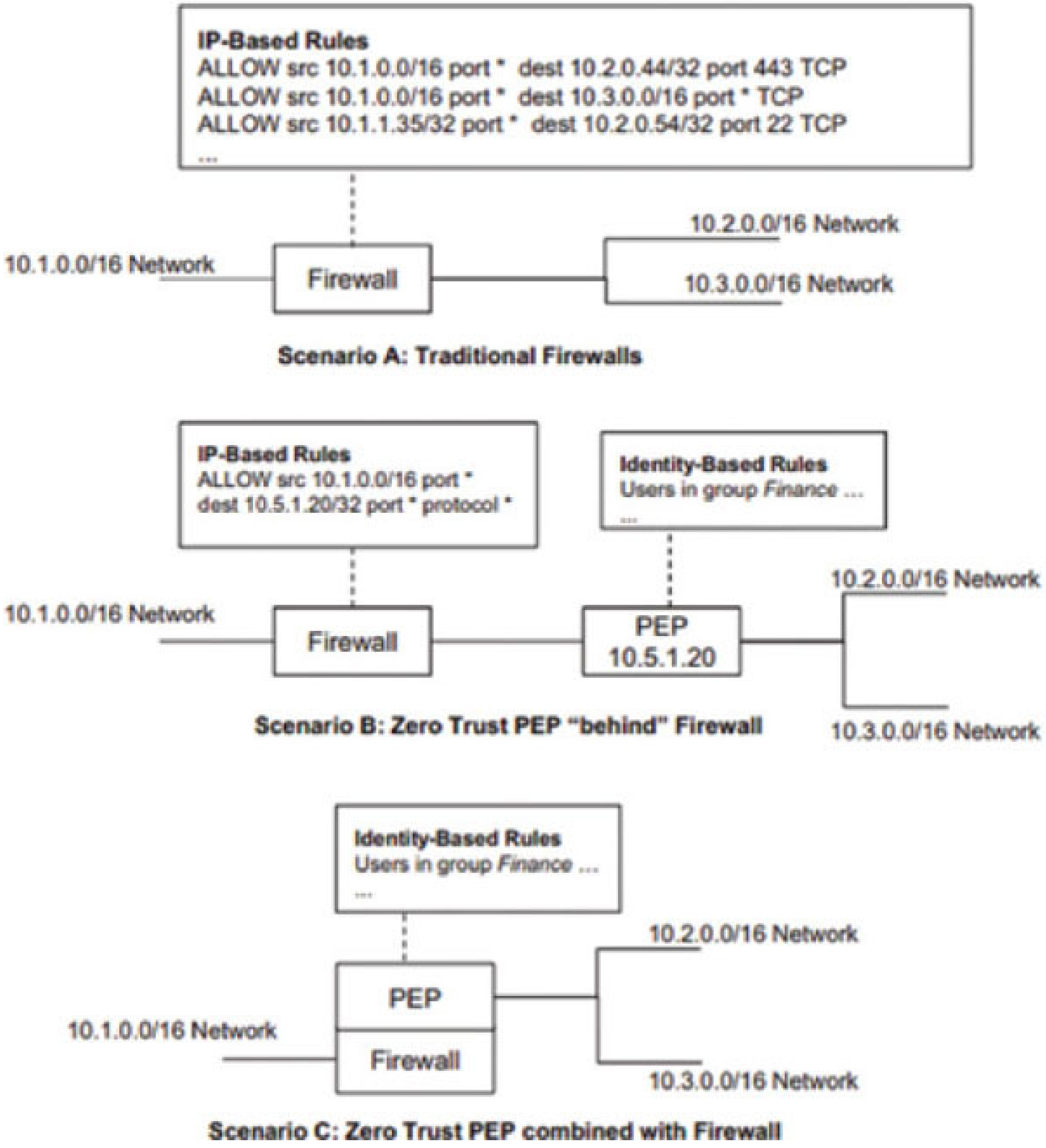

IP-Based Rules
ALLOW src 10.1.0.0/16 port * dest 10.2.0.44/32 port 443 TCP
ALLOW src 10.1.0.0/16 port * dest 10.3.0.0/16 port * TCP
ALLOW src 10.1.1.35/32 port * dest 10.2.0.54/32 port 22 TCP
...
10.1.0.0/16 Network
Firewall
10.2.0.0/16 Network
10.3.0.0/16 Network
Scenario A: Traditional Firewalls
IP-Based Rules
ALLOW src 10.1.0.0/16 port *
dest 10.5.1.20/32 port * protocol *
Identity-Based Rules
Users in group Finance ...
...
10.1.0.0/16 Network
Firewall
PEP
10.5.1.20
10.2.0.0/16 Network
10.3.0.0/16 Network
Scenario B: Zero Trust PEP "behind" Firewall
Identity-Based Rules
Users in group Finance ...
...
10.2.0.0/16 Network
PEP
10.1.0.0/16 Network
Firewall
10.3.0.0/16 Network
Scenario C: Zero Trust PEP combined with Firewall

Figure 6.1: *Firewalls and Zero Trust*

In *figure 6.1*, **Scenario A** depicts a simplified depiction of a typical firewall that enforces IP-centric access restrictions. Traditional firewalls use the basic firewall 5-tuple to define its rules: source IP, source port, destination IP, destination port, and protocol. This narrow (or should we say impoverished) vocabulary only allows access rules to be defined based on (local) IP addresses, not on identities or context, and generally results in overprivileged network access. Of course, this is owing to the fact that IP addresses are neither identities nor unique. The incoming IP address is unlikely to be unique unless the source device and the firewall are on the same network. IP addresses are often translated (remapped and shared) across network or subnet borders, making it hard for the firewall to make identity or context-aware access control choices.

Of course, Zero Trust is intended to address this issue by providing identity and context-aware policy enforcement points on the network. One of two things will happen as a result of this. The first situation, as shown in **Scenario B**, is that the firewall's rules will become much simpler as the PEP essentially takes over. Because the PEP is usually responsible for ending an encrypted tunnel, it is aware of the entity at the tunnel's starting point and can implement identity-centric restrictions.

Scenario C shows an option, in which the PEP is combined with the regular firewall. When the Zero Trust provider is also your (next gen) firewall vendor, this is a possible scenario. Scenarios B and C appear to have basically the same functional consequences on the surface. As you assess distinct suppliers' capabilities in terms of policy models, operations, and management, you'll see discrepancies. Scenario C is the approach used by several **Next-Generation Firewall** (**NGFW**) suppliers, who have introduced Zero Trust PEP capabilities to their firewall stacks in certain circumstances (we'll explore NGFWs in a later chapter).

After all, Zero Trust PEPs are network enforcement points, thus they need to operate as firewalls.

A Zero Trust system will result in simpler firewall setups with fewer rules, as well as a reduction in the time and effort required to monitor and maintain them. Because the enforcement task has been moved from the firewall to the PEP, enterprises may minimize the size, complexity, and expense of their firewalls in some circumstances. That is, the access controls that businesses have sought to implement with firewalls in the past can be accomplished more quickly and efficiently with Zero Trust principles implemented in PEPs.

The Domain Name System (DNS)

The Domain Name System (DNS) is a critical component of our network architecture, as well as a source of frustration (and a popular meme1). DNS stands for Domain Name Mechanism, and it is the system that converts domain names and hostnames to IP addresses, which is how computers interact with one another. DNS does not give any encryption or privacy, and it has some intriguing challenges when users are located remotely.

Public DNS servers

Public DNS has a simple hierarchical structure, with most devices' typical network settings set to query a DNS server on their local area network. This is usually a recursive server that sends uncached requests to a collection of external recursive, root, top-level domain, and authoritative servers.

There are certain security difficulties with public DNS servers, such as confidentiality issues (covered later in this chapter) and issues with DNS infrastructure security (not in scope for our discussion, although there are some promising standards in the works with the IETF). Finally, keep in mind that public DNS entries are (by definition) designed to be accessible to any unauthenticated user on the internet.

Private DNS servers

Private DNS, on the other hand, is a quite different beast, and, as previously said, is the source of much frustration and many jokes. The main reason is that these DNS servers and their information are designed to be private, with only a small number of people having access to them. Part of this anonymity is achieved by having the private DNS server only accessible from a local network, limiting access to only those users who are physically present on that network. Another feature of this anonymity is that private DNS queries typically return private (non-Internet-routable) IP addresses that can only be accessed from that private network. For hostname resolution queries, local devices are often provided a local (private) DNS server as their first starting point. The server will either provide a cached answer or recursively ask its specified external public DNS server, which will return the publicly routable IP address, to resolve public DNS entries. It merely refers to its local database to resolve private DNS entries (for example, server1234.internal.company.com) and returns the server's private IP address.

Most businesses have complicated domains and networks, and DNS complexity grows fast. An business with three internal domains connected by LAN, for example, must ensure that the DNS servers either duplicate their contents or that all DNS servers are available by devices issuing DNS requests.

Internal IP addresses supplied by private DNS servers must, of course, be reachable via the network. We've just talked about local users so far, and things get significantly more difficult when users are located outside of the target servers. With the mix of IaaS-based private resources and numerous users working from home, this is a regular scenario. The necessity for users to access private servers that are located in disparate and isolated domains, such as geographically distant locations or split across IaaS environments, presents a challenge to IT and security teams in these instances.

Traditional remote access systems (read: VPNs) are restricted in their capabilities, often offering either complete tunnelling of all DNS traffic to an internal server or split tunnelling based on domain names (search domains).

All DNS requests are forwarded to the distant DNS server in the first example, whereas some DNS traffic is sent to the local (LAN) DNS server in the second. Of course, zero-trust solutions must address this issue as well, and different platform take different approaches. Some Zero Trust systems demand that internal server broadcast entries in public DNS, directing external users to external (or publicly facing) proxies for the apps. Cloud-based Zero Trust systems frequently employ this technique, which is more static. Other Zero Trust models may use a more complex approach, such as using a PEP based on search domains to send client DNS requests to private DNS servers. This eliminates the requirement for public DNS entries on private servers and allows for dynamic host resolution, which is important in virtual and cloud contexts.

This is a complicated problem, and Zero Trust solutions don't all handle it the same way—heavily it's dependent on the platform design. However, you should ask your potential providers this issue to ensure that it matches with your network design. Based on policy, Zero Trust security systems, in our opinion, should be able to automatically resolve (and make accessible) services operating on private hosts.

Because today's environments are so dynamic, with services being built, changed, and removed on a regular basis, a Zero Trust solution must support these agile, DevOps-style activities rather than causing friction.

DNS security monitoring

DNS traffic monitoring is a common security feature that should be included in your Zero Trust system. DNS queries to resolve known rogue domains are clear evidence of malicious behavior, and should be noticed and handled to as soon as possible by an appropriate enforcement point—this is a high-value and low-risk component of a Zero Trust platform. The basic conclusion is that your Zero Trust architecture should contain DNS monitoring, DNS filtering or blocking, and the ability to quickly respond to known malicious DNS requests by altering user access. If your Zero Trust solution delivers DNS traffic over an encrypted tunnel, be mindful of how and where your company monitors this data, as well as how this may affect it. Before we wrap off our overview of DNS, we'd want to add a little note about how it uses encryption. Standard DNS is an unencrypted protocol that uses UDP; all requests and replies are sent in cleartext, making them accessible to local or intermediate network observers (both malicious and benign). Infrastructure security, on the other hand, is still evolving; there are currently in-development standards (as well as some open-source techniques) that bring encryption into DNS in a variety of ways. The Internet Engineering Task Force (IETF) is spearheading this work, with multiple RFCs addressing various elements of DNS security, including proposed standards for DNS over TLS/DTLS (DoT) and DNS over HTTPS (DoH), both of which encrypt DNS communication. In terms of how they function, both systems differ, while the latter (DNS via HTTPS) is a hot topic in the business for good reason. What's critical for security architects to know is how their security systems use DNS query monitoring or filtering, as well as how a switch to encrypted DNS would affect their visibility and control. Organizations should consider using DoT since it offers security benefits and may be integrated into existing business processes. DNS configurations are made in a non-disruptive manner (although it may impact DNS monitoring as just discussed). Some Zero Trust systems will increase DNS security by design, for example, by allowing DNS requests to be sent over an encrypted tunnel, as previously stated, and controlled by a Zero Trust policy. This would combine the advantages of encrypted DNS with the ability to monitor and filter DNS traffic. Users should not use DNS over HTTPS, according to today's standards, since it circumvents business DNS policies in ways that might be hazardous.

Wide Area Networks (WANs)

Wide Area Networks (**WANs**) have been a mainstay of enterprise networks since the 1980s, connecting geographically distributed enterprise sites and networks long before the Internet became widely available and reliable, with underlying technologies gradually shifting from circuit-switched to packet-switched networks. WANs are more concerned with providing dependable and efficient network connectivity than with ensuring security. In reality, this implies that WAN traffic is often routed privately through carriers or network providers, but without further encryption. That is, while the traffic is not visible to the general public while in transit, it is available to network service providers as well as any other intermediate actor with legal (or illegal) network access. The main conclusion is that network traffic encryption is rarely supplied by the WAN.

Similarly, access control is out of scope for WANs, which are intended to link scattered business networks rather than provide a paradigm for restricting access based on firewall rules or policies. Enterprises that use WANs must, of course, implement and configure network firewalls at the edges, just behind their service provider's WAN router. They must also select whether and how to encrypt traffic passing via the WAN, whether using an encrypted application protocol or another method.

The **Internet Engineering Task Force** (**IETF**) is spearheading this work, with multiple RFCs addressing various elements of DNS security, including proposed standards for DNS over TLS/DTLS (DoT) and DNS over HTTPS3 (DoH), both of which encrypt DNS communication. In terms of how they function, both systems differ, while the latter (DNS via HTTPS) is a hot topic in the business for good reason. What's critical for security architects to know is how their security systems use DNS query monitoring or filtering, as well as how a switch to encrypted DNS would affect their visibility and control. Organizations should consider using DoT since it offers security benefits and may be integrated into existing business processes. DNS configurations are made in a non-disruptive manner (although it may impact DNS monitoring as just discussed). nodes to assure network quality of service, which has ramifications when paired with Zero Trust, which we will cover later. SD-WANs are similar to regular WANs WANs offer network connectivity to dispersed sites but lack a built-in security mechanism or the ability to enforce access regulations

Enterprises may naturally find themselves relying less on WANs (and more on encrypted connections started by user devices) as they build and deploy Zero Trust technologies. This isn't to suggest that WANs will go away; rather, there are two elements that are both contributing to their decline in relevance. First, Zero Trust systems assume the underlying network is unsafe and encrypt all traffic. Second, in today's world, Internet access is mostly ubiquitous, low-cost, and fast and reliable enough to be utilized for mission-critical organizational communications. While most Zero Trust deployments are unlikely to result in rapid changes to WAN infrastructure, they will at the very least open the door to decreasing or replacing them, which is a discussion that network, IT, and security teams should have. Change, by definition, comes with it complexity, and Zero Trust is no exception. Remember that Zero Trust solutions create an encrypted overlay tunnel over intermediary networks, most often between the user agent PEP and the PEP in front of the protected resources. This tunnel is designed to be inaccessible to network intermediates. While this has the advantages of data privacy and integrity, it may also have a detrimental influence on legitimate network intermediates' capacity to execute their responsibilities (a subject that will run throughout this book). In order to satisfy quality-of-service requirements, SD-WANs frequently rely on network traffic information such as port and protocol to make network routing and priority decisions (traffic shaping). This can typically be mitigated to some extent, but it will need collaboration between the Zero Trust and networking teams.

In conclusion, the implementation of Zero Trust will almost certainly have an influence on corporate WAN utilization, frequently to the benefit of the company by lowering expenditures or bandwidth consumption, and in some situations, they may even be abolished.

Because Zero Trust network traffic is generally overlaid on top of existing WANs, you'll need to be aware of how your WAN uses network traffic metadata and how it may be impacted.

API gateways, load balancers, and application delivery controllers

Load balancers, **Application Delivery Controllers** (**ADCs**), and API Gateways are popular networking and IT infrastructure components. They're utilized together to give applications superior speed, scalability, and robustness, as well as create a layer of abstraction between service providers and users. They can be complicated, and they frequently use a combination of technological techniques to achieve their objectives. Even basic load balancers, for example, may allocate workloads to servers using one or more strategies (such as round-robin, random, or load-based). By executing certain network, content optimization, and API aggregation activities in front of their back-end servers, ADCs, and API Gateways minimize server workload.

SSL termination, content caching, connection multiplexing, traffic shaping, and microservices abstraction or consolidation are examples of capabilities. In general, keep in mind that these systems provide network and application activities and are not often thought of as security appliances, except from assisting with availability. The functions offered by these systems are valuable, and they will essentially remain unmodified under Zero Trust systems. However, the possible impact of network topology modifications and new usage of tunneled encrypted traffic within the Zero Trust system should be considered (as with SDWANs, as previously described). This traffic will most likely become inaccessible to intermediate components like these—it all depends on where the PEPs are and how they enforce regulations. In general, load balancers, ADCs, and API Gateways should operate well with the enclave-based and cloud routed deployment models if they are behind a PEP. These components may be hampered by resource-based and microsegmentation models, which may clash with the requirement for an active network intermediate. The key here is to be aware of how your company uses these technologies and to collaborate with your colleagues in networking, application development, IT, and security. Keep in mind that a Zero Trust system does not have to cover all apps and services. It's possible that a web application server (together with a load balancer and ADC) is in place to provide a publicly accessible application that is meant to be visible to and accessible by unauthenticated and anonymous users by design. This may include a company's website or a software-as-a-service application. In contrast, an API service may need to be available from anywhere on the Internet, yet it may benefit from the use of a Zero Trust PEP. It relies on the API access model and the types of client systems that are allowed to use it.

Finally, we'd like to emphasize one more vital aspect.

Even if certain services are intended for usage by the general public and unauthenticated users (such as the website), other services operating on the same host will almost certainly require authentication and authorization and should be included in your Zero Trust scope. A administration interface, such as SSH, will be available on a public-facing web server (or load balancer hardware). This interface's access must be restricted to authorized users and concealed from unauthorized users—a challenge that Zero Trust is ideal for solving.

Web Application Firewalls (WAFs)

Web Application Firewalls (**WAFs**) are security components that sit in front of web servers, processing, monitoring, and securing HTTP traffic in order to protect them. The word WAF is a little confusing because it refers to a security proxy rather than a network firewall. In actuality, these systems are reverse proxies that examine incoming HTTP traffic for SQL injection and cross-site scripting threats and identify and block them.

WAFs are widely used to safeguard public-facing web servers, which are virtually always probed, scanned, and attacked. Such resources clearly justify the investment in security solutions such as WAFs. However, it's worth noting that WAFs are also used to safeguard internal apps that are only available to internal users. They're guarding against malevolent insiders and hacked devices on the internal network in this situation. We commend the increased protection, especially for internal apps, and the presumption of compromise that likely inspired it from a Zero Trust standpoint.

Of course, zero-trust systems can't prevent all assaults, but they can decrease the surface area that a hacked machine can exploit. As an example, a suitable Zero Trust system would limit access to just those individuals with valid business needs and a web app account for a fictional internal web application.

If 10% of the user population utilizes this application, the Zero Trust mechanism will make it impossible for the remaining 90% of devices to even try to attack it. Under terms of WAFs, there's still a need for them in Zero Trust, because 10% of users might be running malicious software that tries to attack the app.

Conclusion

It should be obvious that implementing Zero Trust will have an influence on several aspects of your network infrastructure. Even if your journey has little impact on the network as a whole, all network parts will require some careful consideration and debate. As a Zero Trust architect and leader, you must first learn about your company network and how various security, connectivity, availability, and reliability components are installed.

Zero-trust systems will necessitate this level of coordination, collaboration, and understanding since they serve as an encrypted overlay on top of underlying networks. That's not to imply Zero Trust initiatives won't be disruptive, and we don't want to dissuade you from embarking on this adventure. In truth, there are use cases and circumstances where Zero Trust may be implemented progressively and quickly. However, an enterprise Zero Trust design will have a significant influence on the network and networked applications, necessitating a thorough examination of infrastructure pieces. This chapter, together with the others in Part II, will provide you the background and understanding you need to succeed.

Chapter 7

Network Access Control

For two reasons, we'll discuss **Network Access Control** (**NAC**) solutions independently from the Firewall, DNS, and Load Balancer solutions we discussed in *Chapter 6: Network Infrastructure*. To give credit where credit is due, NAC solutions are early (and ongoing) attempts to implement aspects of the Zero Trust concepts, notably the ability to impose identity-centric access regulations at the network level. Second, as organizations implement a modern Zero Trust architecture, NAC deployments will be impacted—the value and importance of NAC (as an established category) will be diminished, replaced by Zero Trust's more effective ability to serve as a broader and more capable network access control solution.

The basics of NAC

NAC is a term used in the industry to describe a collection of functions and network protocols for identifying and verifying user devices, authenticating users, and enforcing policies that define which network resources users are allowed to access. Device posture checks, such as antivirus protection level, system patch level, and device configuration, are frequently performed by commercial NAC systems. This allows these systems to enforce restrictions such as quarantining defective devices in network segments that are only for cleanup. The machine can access network resources and the Internet after the policy is satisfied, as long as the NAC system's regulations are followed. The intentions of NAC are admirable, and the functions stated are a subset of our Zero Trust principles. So, why are we pessimistic about NAC and consider its future is bleak?

The problem with NAC isn't the aims; it's the way the systems are built. Their strategy (and the network protocol they employ, 802.1x) often necessitates a single company owning and operating the network hardware infrastructure that supports all users and servers. As a result, NAC solutions are useless for distant users with devices linked to a personal or third-party network, or who access cloud-based services.

NAC solutions are hardware-based and do not work in cloud environments or for distant users since they operate at network Layer 2. When utilized for the right scenarios—on-premises users accessing on-premises resources—NAC can be valuable, albeit in reality, it often only provides coarse user assignment to virtual LANs (VLANs), which typically have dozens (if not hundreds) of services accessible. This isn't in line with the Zero Trust objectives. It's also worth noting that NAC solutions don't support network traffic encryption or remote access. There's one more feature common to all NAC implementations that deserves its own discussion: guest network access.

We'll get to that later in this chapter, but first, let's look at 802.1x (pronounced "*eight-oh-two-one-ex*"), the protocol used by all standard NAC systems. 802.1x is an open protocol that provides a network authentication method for authenticating and authorizing devices for access to a LAN. It was established by a mix of IEEE and IETF documents. In a nutshell, a NAC system authorizes a device's access to a network and allows (or prevents) it from obtaining an IP address within that LAN. As seen in *figure 7.1*, this operates in combination with network hardware (switches):

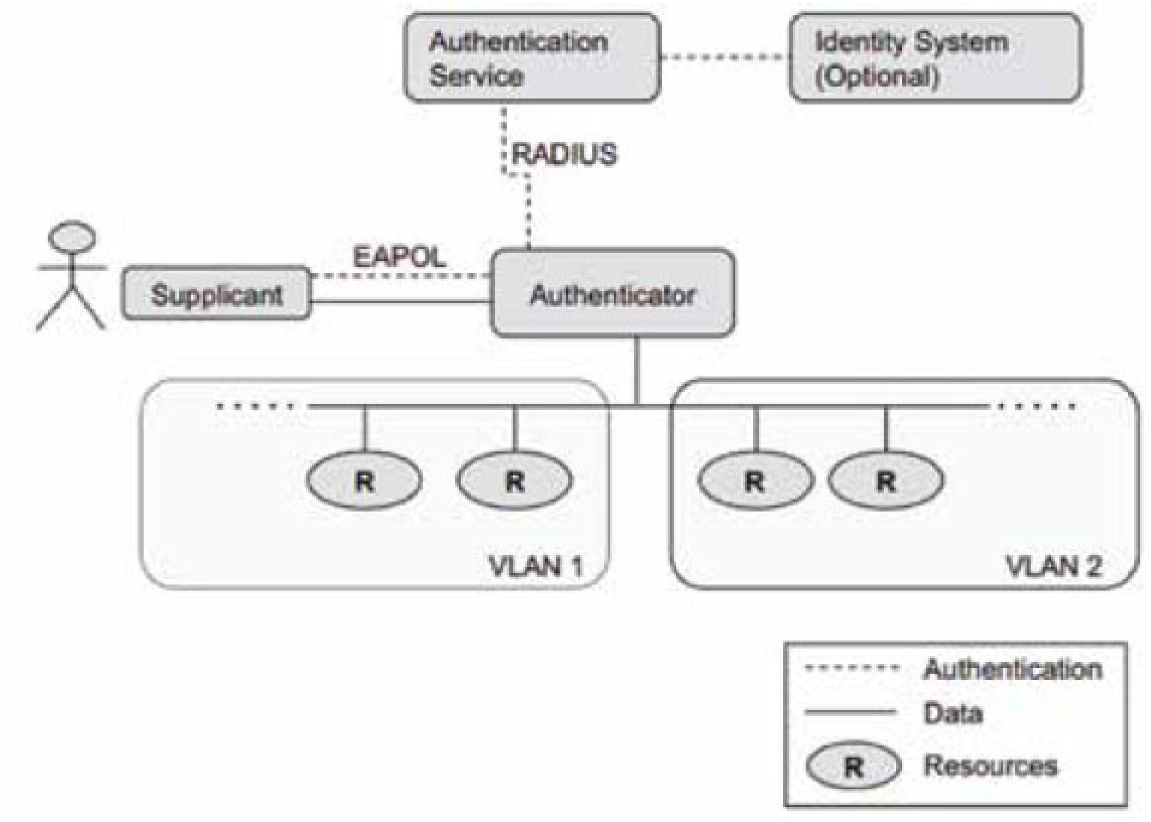


Figure 7.1: *802.1x Authentication*

As shown in the diagram, the user's device (referred to as the supplicant) communicates credentials or certificate information to the Authenticator via the **Extensible Authentication Protocol** (**EAP**) over LAN (EAPOL). When the supplicant connects to the network switch for the first time, it is configured to "*unauthorized*," allowing only EAP traffic—UDP, TCP, and ICMP are not allowed.

EAP is a relatively simple protocol that works at layer 2 following the IP layer by design. As a result, it's essentially a local network protocol that can only be accessed on the local subnet (broadcast domain) and cannot be routed.

The Authenticator communicates with the Authentication Service to verify the user's credentials, usually over the RADIUS protocol. User credentials verified against an identity system or certificate-based authentication may be supported by 802.1x products.

If the user's credentials are legitimate, the Authenticator switches the network switch on the supplicant to "authorized," allowing the device to get an IP address and begin delivering UDP, ICMP, and TCP traffic. Most significantly, the Authenticator makes a configuration setting on the network switch that allocates the device to a network segment (a virtual LAN, or VLAN).

This protocol implies that the supplicant and authenticator both be in the same network broadcast domain—that is, they must both be using the same physical network medium (Ethernet or Wi-Fi). They must also employ network gear that is owned and controlled by the company, and this hardware must be widely distributed across the infrastructure.

As a result, NAC is useless for distant users or those who need to access cloud services. In each of these cases, the user or the service they're accessing (or both) is operating on network infrastructure that isn't owned by the company. As a result, these scenarios, which are becoming more prevalent, pose a substantial constraint in NAC's efficacy.

Many NAC systems don't become involved in access control after a user's device is allocated to a VLAN (other than a periodic reauthentication). The organization's firewalls (or next-gen firewalls) are then responsible for controlling user or device access within the VLAN, which, of course, have their own access policy model. It's worth noting that some sophisticated NAC suppliers offer extra features (not covered by 802.1x) and facilitate interaction with other security components.

Finally, 802.1x only allows for coarse-grained user assignment to virtual LAN segments (VLANs), which are typically used when there are dozens, if not hundreds, of services or peer devices accessible on the network. NACs further exacerbate the problem of excessively broad network access because each device can only be allocated to a single VLAN at a time. While firewall ACLs may be used to supplement NACs, they tend to be static and IP-centric, and hence aren't well aligned with our Zero Trust principles.

Zero Trust and NAC

The notion of least privilege is incompatible with the coarse-grained assignment of devices to networks, which in reality allows users broad network access to all ports and protocols throughout an entire VLAN. This isn't to imply that a NAC solution can't be used in a Zero Trust environment; in fact, we'll look at a few examples later, and NAC's position in the BeyondCorp architecture was previously outlined. However, before selecting how to use NAC vendor solutions in your Zero Trust architecture, you should thoroughly examine how they translate to your Zero Trust needs. NAC providers are well aware of the limitations of the 802.1x protocol, and some have expanded their product portfolios to include features outside the standard to address these restrictions. Some NAC manufacturers, for example, offer endpoint inspection and remote access capabilities, or even offer NAC as a cloud-based service.

Also keep in mind that a significant number of devices in workplace networks, such as printers, VOIP phones, and IoT devices, do not support 802.1x. VLAN assignment is commonly based on MAC addresses in NAC solutions, however these techniques are confined to local network access restrictions and are sometimes difficult to administer. In Chapter 16, we'll look at how these devices are treated in Zero Trust situations.

In any case, some NAC functions, such as guest network access and device detection, make sense to keep even with a Zero Trust architecture, especially if they're already in place. Let's have a look at what they have to offer.

Guest network access that isn't managed

Guest network access is one area where, in some ways, a Zero Trust network can make it less of a problem. Let's start by defining this so that we all know what we're talking about when we do our analysis. The procedure and control of allowing Internet connectivity to non-employee users using unmanaged devices is known as guest networking. The guest network offers Internet connectivity and may contain a few other devices that are only available to visitors, such as wireless conference room A/V systems or a guest printer, but these users and equipment must be kept separate from the enterprise's staff network.

It's worth noting that currently, guest networks are nearly all wireless (Wi-Fi) rather than wired, therefore our talk will focus solely on these sorts of networks. Many (or even most) guest networks are secured with a static Wi-Fi password and consist of a single flat network segment, allowing all devices on the network to communicate with one another. This is adequate security in some environments—a it's reasonable technique when, for example, the guest network **Wireless Access Point** (**WAP**) is separated from the corporate network and there's no concern about visitors accessing critical assets or engaging in harmful conduct. Enterprises can opt to run this guest network with minimal monitoring and control, or they can invest in these capabilities. The important thing to remember is that there is no user or device authentication in this technique by definition—all users are considered the same, and no attempt is made to differentiate between users or device types. In the section *Managed vs. Unmanaged Guest Networks: A Debate*, we go into the ramifications of this strategy. Finally, remember that the 802.1x protocol is not used by unmanaged guest network access, which is a frequent feature of WAP devices.

Managed guest network access

Managed guest network access is a feature included in many commercial NAC systems, and it often includes the following features:

- Log in to the registration portal, which is usually done by email or SMS verification.
- Basic procedure for employee (sponsor) requests for guest access, including deployment of temporary network access

The main difference between unmanaged and managed guest network access is that the latter involves user self-identification and authentication, and usually only provides access for a limited duration. In most cases, these systems require visitor registration via a basic portal (either self-service or by a sponsor employee) and only provide access for a short time (typically 24 hours or less).

Beyond the fundamentally proximity-based nature of short-range Wi-Fi, time-limiting access adds an extra degree of protection. Commercial NAC systems often include a controlled guest network portal and workflow.

A debate on managed vs. unmanaged guest networks

While not as heated as the Lincoln-Douglas debates, there are a variety of opinions and methods to guest network security, with no obvious right or wrong answer—organizations must make their own judgments depending on their environment and risk profile. As indicated in *table 7.1*, there are numerous capabilities that can be associated with a network that businesses must consider. These capabilities can be loosely classified as less secure to more secure. These are fascinating trade-offs that hold true even in Zero Trust networks. It's important to remember that a guest network must always be kept distinct from the employee or corporate LAN:

Network Security	Attributes
Open Wi-Fi: No Password	There is no encryption of network communication. There is no authentication or identity of the user.
Password-Protected Wi-Fi	Encryption of network traffic There is no authentication or identity of the user. Accepting Terms of Service through a captive gateway (optional)
User Registration	Access to the network is restricted in time. Self-identification and authentication of users (usually by email, not against a directory). Acceptance of Terms of Service Form
Employee Sponsorship	Access to the network is restricted in time. To gain access, you'll need to have a verified employee workflow. Authentication and identification of users Acceptance form for the Terms of Service.
Device Isolation	Even if these devices are linked to the same Wireless Access Point, certain Wi-Fi networks allow you to separate them from one another using router firewall rules. This is a good practise to prevent curious individuals (or viruses) from scanning the local network for network and port scans.
Network Monitoring	DNS filtering and IDS/IPS are examples of services that are routinely utilized on business networks.

Table 7.1: *Network Security Attributes*

Each business and security team must make their own conclusions, but we believe that a password-protected Wi-Fi guest network, as long as it is distinct from the corporate network, is likely sufficient for most enterprise contexts.

If WPA3 is available, it is preferable over WPA2, and device isolation is a pleasant bonus but can be regarded optional. Enterprises using Zero Trust networks should, of course, continue to offer guest Wi-Fi with the right combination of network security properties for their environment. The necessity for a guest network is unaffected by a Zero Trust network, as the previously described issues remain unchanged.

One intriguing side note: Your company's guest network will almost certainly be as secure as public Wi-Fi networks like those found in airports and coffee shops. Your Zero Trust solution must also allow your users to access business resources from such networks, as we've explored throughout this book. As a result, there's no reason why your regular employees can't utilize the guest network as well, just as if they were working remotely. Enterprises, on the other hand, usually have extra security or compliance restrictions, as well as greater bandwidth, for their employee networks. As a result, they may prefer that staff utilize the employee network on a regular basis rather than the guest network.

Employee BYOD

Many companies allow employees to access corporate networks and controlled resources using personal devices. This might involve the usage of a personal smartphone or tablet, as well as a user's choice for a certain laptop model or operating system. Organizations can take a hands-off approach, allowing users to log in from any device, or they can impose some level of business footprint, such as the installation of enterprise-issued certificates or device management software. Employees can use BYODs to access company resources, but security teams must decide if and how they can do so. This may or may not be possible with standard NACs, depending on how strict network access is restricted and how much the security team demands the installation of certificates or management software on user devices. The *table 7.2* summarizes the various techniques. Note that this is typically consistent for both laptops and mobile devices, while there may be slight changes across operating systems and security platforms:

Device Configuration	With NAC	With Zero Trust
Pure (BYOD)—nothing is installed or set.	Internet connectivity is available on the guest network. Wi-Fi password security allows	Internet connectivity is available on the guest network. Wi-Fi password security allows for

	for coarse-grained (full network) access limits. Only on-premises users are affected.	coarse-grained (full network) access limits. Only on-premises users are affected.
(BYOD) with a corporate certificate installed	Access to the employee network (VLAN) through the built-in 802. 1 supplicant, 1 supplicant, 1 supplicant, 1 suppl Provides access restrictions with a coarser granularity. Only on-premises users are affected.	“Pure BYOD” is the same thing. Access to the device certificate storage usually necessitates the installation of client-side software.
BYOD (bring your own device) with software loaded and setup (company certificate optional).	802.1x (either built-in or added) access to the employee network (VLAN). Device posture checks can be performed using installed management software.	Zero Trust client installation allows for granular network access management. For conditional access, you can employ certificate and device posture checks. Both on-premises and remote users are affected.

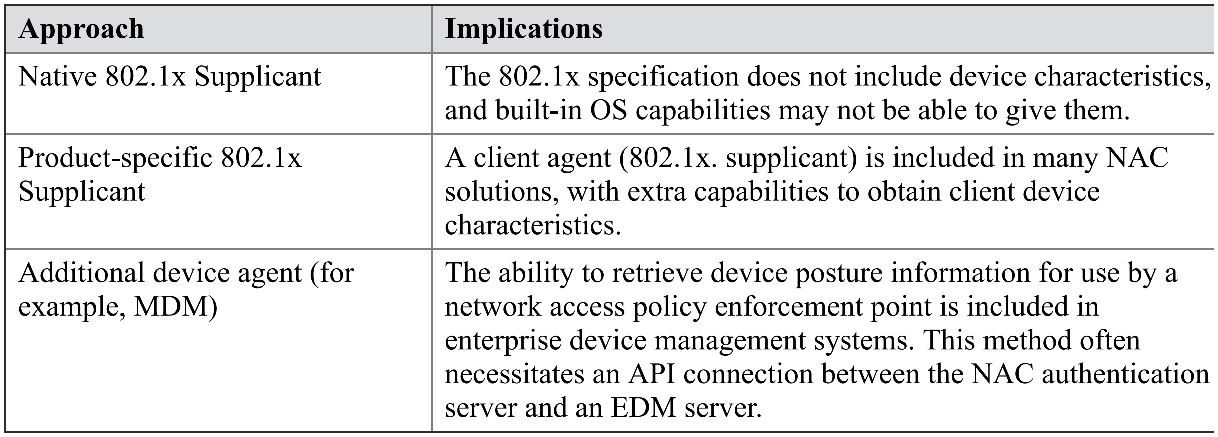
Table 7.2: *BYOD Configuration Comparison*

Device posture checks

In the end, this set of requirements—blocking all access by unauthorized users and/or devices, quarantining/limiting access by authorized but non-compliant devices, and allowing limited access by authorized users on validated devices—are common to both NAC and Zero Trust solutions, and are important security goals regardless of the implementation. We’ve attempted to illustrate how 802.1x-based NAC solutions function throughout this chapter, as well as their weaknesses in terms of adhering to Zero Trust principles. Following that, we’ll look at device configuration analysis. This feature is not covered by the 802.1x standard, although it is widely seen in NAC solutions. NAC systems frequently include the ability to do user device configuration checks, also known as posture checks. This combines the ability to retrieve device information—such as the level of OS patching or the presence of an up-to-date antivirus solution—with the ability to define and enforce a policy that determines which network resources (if any) a device should be able to access based on its device profile. If a device is not “up to date” with security or anti-virus patches, confine it to an “*IT*

Remediation” VLAN with access to only the IT helpdesk portal/self-service portal.

This is a desirable aim; in fact, any Zero Trust policy and enforcement model must incorporate device properties. Of course, this necessitates the capacity to gather device data for use in the policy—the various approaches are given in *table 7.3*:



Approach	Implications
Native 802.1x Supplicant	The 802.1x specification does not include device characteristics, and built-in OS capabilities may not be able to give them.
Product-specific 802.1x Supplicant	A client agent (802.1x. supplicant) is included in many NAC solutions, with extra capabilities to obtain client device characteristics.
Additional device agent (for example, MDM)	The ability to retrieve device posture information for use by a network access policy enforcement point is included in enterprise device management systems. This method often necessitates an API connection between the NAC authentication server and an EDM server.

Table 7.3: *Device Posture Approaches with NAC*

In the end, being able to get device characteristics is only one component of the equation—and, we’d say, a minor one. What’s most essential is the ability to define and implement a dynamic policy model to regulate network access based on these properties.

In *Chapter 17: A Policy of Zero Trust*, we’ll go through a Zero Trust policy model in further detail.

Device discovery and access controls

Understanding what’s on a network (including people, devices, and workloads) is, of course, required before implementing a policy model and imposing access rules. Entities must be authorized, and the system employs a number of factors to make contextual access choices, according to any Zero Trust paradigm.

NAC systems can be used as part of a Zero Trust solution (for both coarse-grained network assignment and device discovery information, such as in BeyondCorp), but they cannot offer dynamic, fine-grained, and universal access restrictions for all users and resources.

The *table 7.4* summarizes and compares NAC and Zero Trust techniques in terms of what security and network teams must do on corporate (nonguest) networks:

Device type	With NAC	With Zero Trust
Unauthorized Devices	Block all network access: no VLAN, no allocated IP address	Access to any protected resources is blocked. Access to the internet may be restricted. The device is connected to the internet, but it is unable to access anything.
Authorized but unmanaged or non-802.1x Devices	MAC address grouping is commonly used to assign VLANs.	Controls for access depending on device type (for example, MAC address grouping). VLAN is finer grained than this.
Authorized and managed or 802.1x-enabled Devices	Authenticate devices and assign them to a VLAN (based on identity groups, for example).	Apply fine-grained and identity-specific access controls after authenticating.

Table 7.4: *Device Security Approaches on Enterprise Networks*

One final point: utilizing device MAC addresses for access restrictions is, of course, a "*not great but better than nothing*" approach, and any use of this method must be accompanied by a thorough knowledge of the risks and threat model involved. A hostile actor with physical access to a network may easily modify their MAC address and impersonate an authorized device to get access. Knowing this, it's critical to run through the thought experiment and make sure that if something like this happened, the device would only have extremely restricted access (for example, to a printer or VOIP VLAN).

Conclusion

We covered the basics of Network Access Control in this chapter, as well as how the 802.1x protocol works. We next looked at NAC solutions from a Zero Trust viewpoint, examining how NAC solutions handle guest network access, which is still a requirement in Zero Trust networks. Finally, we looked at BYOD, device profiles, and device detection as well as other facets of NAC.

NAC solutions can be used in a Zero Trust environment, particularly if they include essential capabilities for guest network access restrictions, although 802.1x-based NAC functions are not suited for usage as the heart of any Zero Trust environment. Some NAC suppliers have gone beyond 802.1x and introduced Zero Trust features, but companies adopting them should compare them to their network and architectural needs carefully.

Chapter 8

Intrusion Detection and Prevention Systems

Intrusions, which we'll define as unauthorized software execution or unwelcome human behavior on an enterprise device or network, are obviously a requirement for enterprise security solutions. **Intrusion Detection Systems** (**IDS**) can detect, log, and notify on suspicious behavior, whereas **Intrusion Prevention Systems** (**IPS**) can respond by blocking or terminating the activity in some way. To detect undesired activity, Intrusion Detection and Prevention Systems (IDPS) often use signatures (pattern matching) and/or anomaly-detection procedures (sometimes employing statistical analysis or machine learning). In order to receive current data to educate their algorithms, they frequently integrate with threat intelligence systems. These products are widely accessible as independent products as well as integrated into various **Next-Generation Firewalls** (**NGFWs**).

IDPS have historically been most effective in circumstances where they can be deployed into well-defined network *chokepoints*, get insight into traffic, and, ideally, do deep packet inspection. PEPs are a logical place for certain activities to be performed in a Zero Trust architecture. In fact, we think that when incorporated into a Zero Trust system, current IDPS may effectively become a PEP.

When IDPS can consume and enforce Zero Trust policies, as well as operate as a source of events to the PDP, which will in turn prompt other PEPs to take action and, for example, start changes in user risk levels or access, they will increase their value and efficacy. One additional point of clarification: we've just explored network-based IDPS so far; there's also a type of host-based IDPS. Next, we'll compare them to see what they do and how the transition to Zero Trust affects them.

IDPS varieties

As illustrated in *table 8.1*, there are two broad methods to commercial IDPS: host-based and network-based, which differ in where and how they're installed. It's worth noting that, in general, preventive systems must have at least some detection features; in order to intervene, they must first notice undesired (or at the very least unexpected) behavior:

Function type	Detection	Prevention
Host-Based	Process behavior analysis File integrity monitoring. Metadata analysis on networks Analysis of local logs and events Forwarding of logs and events Monitoring of device or user behavior Monitoring of software installation or downloads Detection of privilege escalation or rootkits	Process whitelisting Process termination Software download or installation prevention Termination of network connections
Network-Based	Network metadata analysis DNS monitoring Deep packet inspection (network traffic inspection)	DNS filtering Network content blocking Network connection termination Suspicious content "detonation" in a sandbox

Table 8.1: *Typical Intrusion Detection and Prevention Functions, by Deployment Model*

The table 8.1 demonstrates that security solutions include a wide range of potential roles and procedures for detecting and responding to unexpected activity. One of the reasons why modern IT and information security may be so difficult is that there are so many different ways for malicious actions to manifest themselves, and so many different ways to guard against them. Some of these activities, like as terminating a network connection, plainly fall under the purview of a Zero Trust system. Others, such as host-based process termination or "detonation" of sandbox payloads, are likely outside the reach of a Zero Trust system. Nonetheless, those systems can bring value to a Zero Trust platform by serving as a source of data or events.

Let's look at the two IDPS deployment options and explore the consequences of each as an organization goes to a Zero Trust architecture now that we've established some context.

Host-based systems

A software agent operating on either user devices or servers is used in host-based intrusion detection and prevention systems (resources). Host-based systems have the benefit of being able to analyze everything that happens in the OS in detail, including process and network activities, as well as execute local actions. Many Zero Trust implementations, which often use encrypted traffic tunnels across networks, benefit from having a local presence on the host (we explore this further later in this chapter).

One drawback is that these systems necessitate the installation and administration of software on potentially vast numbers of devices, and they frequently require higher rights to operate. These agents may also shorten the battery life of mobile devices, interfere with genuine user activities, and degrade device performance across all platforms.

This last component might be a source of worry in terms of how it affects the end-user experience.

Having said that, we do suggest that businesses put some kind of agent on devices that access protected resources for a variety of reasons, as detailed in *Chapter 3: Architectures With Zero Trust*. However, one issue that IT teams frequently confront, particularly on end user devices, is agent proliferation (and functional overlap amongst agents). Because these agents are often delivered as binaries with their own unique deployment footprint, dependencies, and settings, this is an unresolved challenge.

Agent consolidation and vendor release synchronization are unlikely to occur, and companies should not expect this to happen. Instead, security teams must accept reality and approach agent deployment with caution.

Fortunately, current devices and operating systems often have sufficient memory and processing power to enable the operation of numerous agents without difficulty.

Network-based systems

IDS and IPS that are network-based are installed in an organization's network and have the capacity to monitor (and perhaps change) network traffic. Of course, modern networks are dispersed and segmented, and the breadth and capabilities of any network-based IDS/IPS system is fully reliant on the location of the system nodes and the sort of network traffic they can access. An IDS implemented within an employee LAN subnet, for example, might possibly investigate traffic between devices on that network, as well as traffic between those devices and distant resources. An IDS on a WAN link connecting scattered data centers might be able to investigate inter-data center traffic, but it won't be able to observe any local LAN traffic within a data center or network. A network tap or span port (passively watching traffic) or an in-line IDPS can be used to implement network IDPS (observing traffic as it transits through the node). The latter method has the benefit of being able to terminate connections more reliably in the event of an identified danger. One of the reasons that NGFWs, which integrate intrusion detection and prevention as a feature, are a popular and still-growing market segment is because of the in-line approach.

One may argue that an organization's ideal network based IDPS deployment would have nodes *everywhere*, allowing the system to monitor "*all*" network traffic in aggregate. Our counterargument, on the other hand, is:

- Capital and operating expenditures may restrict organizations' ability to install network based IDPS throughout the whole network.
- Many users and resources in today's environment operate on third-party networks, outside of the control of the enterprise—with employees working from home or on hotel networks, accessing resources that may be hosted in the cloud (especially IaaS).
- Finally, the widespread use of encrypted network protocols makes it more difficult for network based IDPS to be as effective as they have been in the past.

Because the encrypted tunnels usually used by Zero Trust systems render traffic mostly opaque to network intermediates, this final bullet point is significantly more essential in Zero Trust contexts, and frequently makes network-based IDS more difficult to install. The influence of encrypted network protocols on network-based security solutions is a fascinating subject that we'll discuss next.

Encryption and network traffic analysis

Clearly, a Zero Trust solution alters the security architecture and network of a company.

It will alter how different IT and security components interact with one another, as well as the network itself—possibly from a topological standpoint, but most likely via modifying network segmentation and imposing extra levels of encryption. This final modification is extremely crucial to comprehend, especially for firms that use network-based IDS solutions.

Because encryption (most frequently TLS) preserves message integrity and secrecy from network peers or intermediaries, modern application protocols should use it. Of course, this means that even authorized network intermediates seeking to execute security functions are unable to see the contents of encrypted communication.

The well-established practice of having the intermediary be an active participant in the discussion, stopping one encrypted link and commencing the other to undertake traffic inspection, is the answer to this problem. This is usually accomplished by disseminating a business PKI-generated certificate, which is based on a root of trust shared by both sides of the encrypted TLS connection—basically, a lawful **Man-In-The-Middle** (**MITM**) attack.

We noted that this method is well-established, and many security solutions utilize it to inspect encrypted communications. It is, however, based on a basic use model: one-way authentication with a static set of certificates, in which the client authenticates the server's certificate as part of establishing the TLS connection, but the server does not do any network-level validation of the client.

This is particularly helpful in models where the server should lawfully accept connections from any client and then do client authentication at the application level later in the process. Because it only involves the distribution of a single, static server certificate, this architecture makes it possible for an intermediary network security component to conduct TLS termination.

In Zero Trust situations, however, this concept may not be appropriate.

For communications between the user agent PEP and the network PEP, many Zero Trust implementations employ mutual TLS (mTLS, also known as two-way TLS). Both PEPs validate each other's certificate using this way. This improves security since a hostile actor can't launch an MITM attack with only one stolen certificate—they'll need both components' certificates, which is a far less likely situation. For these communications, some Zero Trust systems go even farther and employ short-lived certificates.

As a result of the increased security, a normal network IDPS operating "between" PEPs will be unable to access encrypted network traffic.

That is, the IDPS will not have access to the certificates used to encrypt the Zero Trust tunnel, even if it has access to the certificates used to encrypt the application protocol.

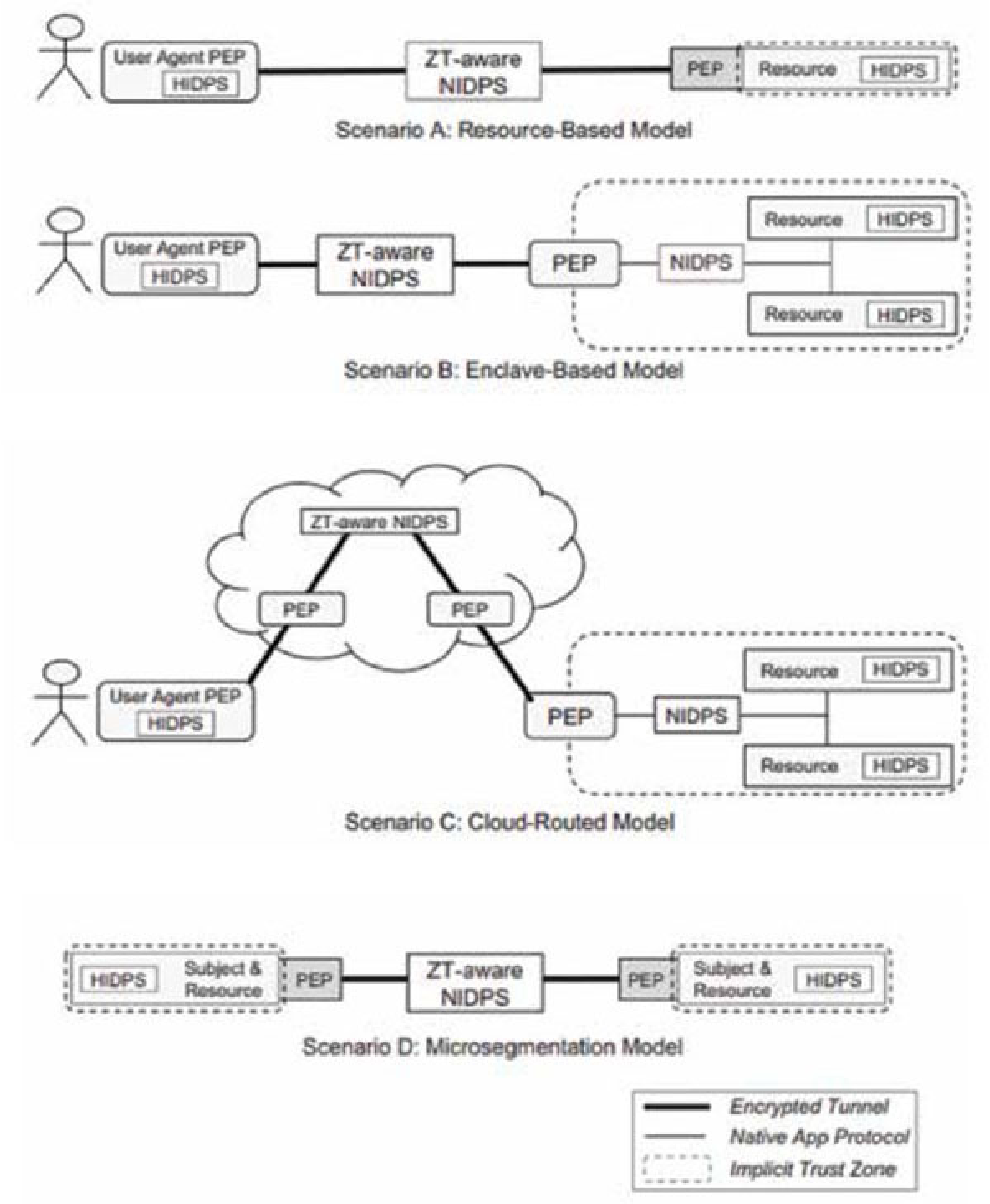
We'll go over this in more detail in the following section, but first, a quick remark about TLS. TLS v1.3 (finalized in August 2018) is being phased in by the industry, which alters several security components of the TLS connection and makes network security solutions more complex. Additional elements of the TLS handshake have been encrypted, making it more difficult for a passive network monitor to identify malicious activities. We recommend this comprehensive and well written IETF document, Impact of TLS 1.3 on operational network security practices, if you want a more in-depth look.

The bottom line is that the transition to TLS 1.3 is underway and should be welcomed rather than resisted.

Zero Trust and IDPS

Modern security architecture necessitates IDPS, which is described as a general collection of functions that spans an organization's platforms in its broadest meaning. Even as enterprises progress toward a Zero Trust security architecture, this will remain critical. With the implementation of a Zero Trust strategy, however, the how of IDS/IPS is likely to alter.

Organizations must be aware of this and willing to adapt as a result. Network segmentation and network traffic encryption patterns, for example, will alter as a result of Zero Trust. This may need a greater usage of host-based IDS/IPS or a greater investment in network-based IDPS as part of a Zero Trust system. Figure 8-1 illustrates this:



Scenario A: Resource-Based Model

Scenario B: Enclave-Based Model

Scenario C: Cloud-Routed Model

Scenario D: Microsegmentation Model

Figure 8.1: *IDPS and Zero Trust Deployment Models*

The *figure 8.1* depicts the four Zero Trust deployment approaches, including network-based IDPS (NIDPS) and host-based IDPS (HIDPS), as well as encrypted (tunneled) network traffic and implicit trust zones. The NIDPS may be blinded by tunneled traffic, depending on the Zero Trust deployment strategy. This is evident in all of the cases in *figure 8.1*, where an NIDPS must be "Zero Trust-aware" in order to function across PEPs—that is, it must be a member of the Zero Trust system and be able to decode tunneled communication. Standard NIDPS can continue to function, but only in situations B and C, where a network segment within the implicit trust zone may be used to install the NIDPS. Because they run on hosts and so have access to network traffic "*behind*" the PEPs, host based IDPS will be mostly impacted by the encrypted network traffic. While host-based systems can continue to operate as before in a Zero Trust environment, they can actually give additional value by being even loosely linked. If the Zero Trust system suggests a greater risk score for the host's network, for example, a host-based system on a server may alter its level of examination and alerting.

Rather than being a separate tool, IDPS capabilities are more likely to be "baked in" to an enterprise's Zero Trust platform.

You may argue that the Zero Trust system will eventually become the IDPS, depending on its capabilities. That is, IDPS ceases to be a discrete function and instead becomes an integral element of the overall security architecture. They can be accomplished by PEPs that have IDPS capabilities, or by standalone IDPS that sit "behind" the PEPs and are integrated into the Zero Trust environment to some extent. They should be able to modify the amount of inspection and enforcement by consuming regulations, resource information, or identity context.

For example, network traffic passing through an IDPS accessing a low-value resource may not require as thorough (read: resource-intensive) analysis as traffic accessing a high-value resource.

Access from a local user on a corporate-managed device, for example, may be subject to less scrutiny than access from a distant user on a BYOD device.

These kinds of integrations can lower the amount of infrastructure needed to support the IDPS, as well as the number of false positives that IDPS are prone to.

Integration with Zero Trust also allows the IDPS to respond to suspected intrusions with a larger range of actions. While IDS and IPS can only detect and restrict attempted network access, Zero Trust solutions have a far larger reach and can take global action. They can, for example, request step-up authentication from users or quarantine user devices across all networks.

Let's look at another area: client-side security solutions (which typically include antivirus and IDPS in one package) are an important part of a Zero Trust security architecture. However, when combined with a Zero Trust policy model that can operate as a network enforcement point, these solutions can provide better security (and more value).

For example, before granting access to enterprise-managed resources, businesses may wish to set an access policy that needs up-to-date antivirus signatures on client devices. The data for the client profile might come from the client slide or a central antiviral management system.

In any instance, the Zero Trust system, which acts as a network-based Policy Enforcement Point, can prevent non-compliant devices from accessing resources and, for example, restrict access to an IT helpdesk or self-service system for updating antivirus signatures. This policy may be implemented independently of the user's location or the kind and location of resources they are accessing since Zero Trust controls all network access to all business resources.

These are examples of how we feel these capabilities should be viewed—not simply as IDS or IPS functions, but as sources of data (input) into Zero Trust systems, as possible catalysts for Zero Trust system action, and as mechanisms for policy enforcement.

By harmonizing policy enforcement across the network and eliminating superfluous or duplicate enforcement points, this approach can enhance both security and efficiency.

There is no one-size-fits-all solution to deploying these capabilities; it is totally contingent on each company's security architecture, ecosystem, and Zero Trust strategy. With so many different goods and so few standardized ways to connect them, this is a difficult challenge to solve. The good news is that progress is being made in this area by the industry. Through the STIX and TAXII standards, for example, the threat intelligence community has been creating and advocating standardized and organized ways to encode and convey threat intelligence information.

Consider a standards-based threat intelligence feed that alerts a Zero Trust system to newly discovered malware that targets specified application types and exploits a vulnerability in a specific desktop client OS version. This information may be used to improve the level of inspection an IDPS applies to the targeted apps, as well as to activate the Zero Trust PEP, which requires the installation of a client OS patch before any access is permitted.

We're excited about the kind of connections that these standards will enable, since they will help enterprises get more value out of their existing IT and security infrastructures and make faster progress on their Zero Trust journeys.

Conclusion

The ideas underpinning Intrusion Detection and Prevention Systems, as well as the set of functions that these systems normally do, were explained in this chapter. We also explored the influence of encrypted network protocols on IDPS and contrasted the two basic forms of IDPS, host-based and network-based IDPS. Finally, we examined the possibility for IDPS to serve as a Zero Trust Policy Enforcement Point from the standpoint of Zero Trust deployment models.

CHAPTER 9

Virtual Private Networks

Virtual private networks (**VPNs**) were originally developed and deployed in the mid-1990s in response to the growing acceptance of workplace networks, as well as the widespread use of PCs at home (either “portable” or “desktop” PCs). Of course, the underlying network protocols have evolved and grown more standardized (and secure) over time, but the essential notion has not:

Between remote nodes, an encrypted network tunnel is formed, allowing application traffic to be safely delivered through an untrusted intermediate network.

As shown in *figure 9.1*, the word “VPN” is now overused, referring to three different types of solutions:

- **Consumer VPN**: Provides privacy and security by shielding end-user Internet-bound communications from intermediaries. Frequently used to protect privacy or get around ISP or government limitations.
- **Enterprise VPN**: Establishing a secure connection between remote users and an organization’s network. This is our primary emphasis, as well as the VPN type most affected by Zero Trust.
- **Site-to-site VPN**: This is one method of establishing WANs for businesses.

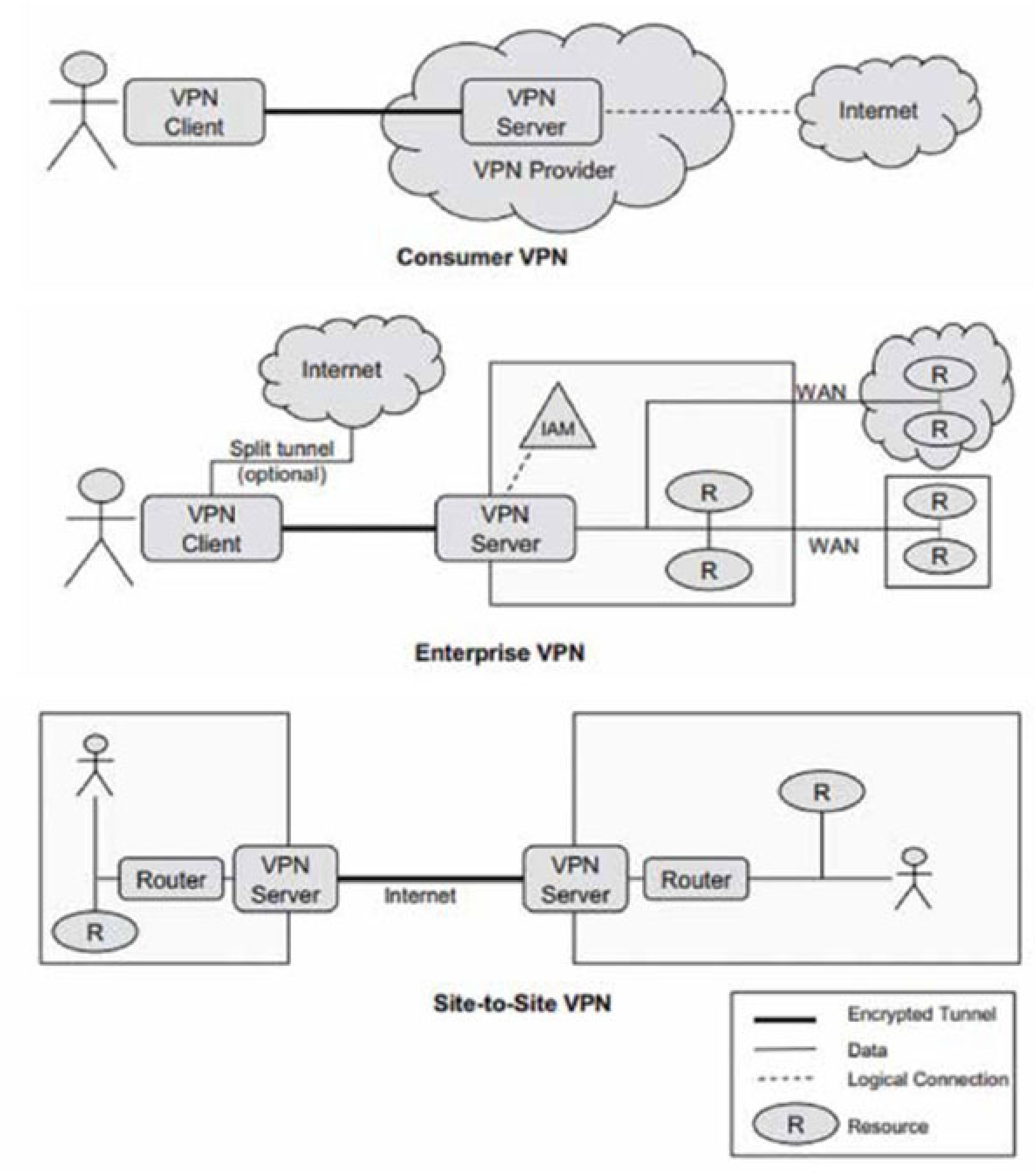


Figure 9.1: *VPN Types*

In order to construct a secure encrypted tunnel, VPNs require two collaborating components that use a shared secret and/or a shared root of trust1. The tunnel between a user's service running a VPN client and the VPN Server is established by user-centered and business VPNs (sometimes referred to as the VPN Concentrator or VPN Gateway). In this situation, the

VPN client might be standalone software, built into the user’s operating system, or even operate within a browser.

To preserve privacy and integrity across the untrusted intermediate networks, part or all user traffic is transmitted over the encrypted tunnel in both of these instances. The traffic is extracted from the encapsulating tunnel and sent to its destination once it reaches the VPN server.

The traffic is extracted from the encapsulating tunnel and sent to its intended destination once it reaches the VPN server. Consumer VPN traffic is sent to an Internet destination, whereas Enterprise VPN traffic is routed to an internal corporate network destination. Many corporate VPNs allow for “split tunnelling,” which means that only traffic headed for the company network is transmitted over the tunnel; all other traffic is sent straight from the user’s device. The complete tunnel option transfers all of the user’s traffic to the company. Although this increases latency (and bandwidth costs), it allows the organization to execute security operations on all of the user’s communications. Site-to-site VPNs function a little differently in that they create a secure encrypted WAN tunnel between two fixed sites, thereby converting them into one logical LAN. In this instance, part of the traffic on those LANs will be routed over the VPN link in order to reach the distant destination. Users are completely unaware of this; they are not using any VPN software, and their traffic simply reaches its destination.

Enterprise VPNs and security

Let’s take a look at the Enterprise VPN scenario, starting with the benefits they bring. First and foremost, they offer an encrypted tunnel between the user’s device and the company network for user traffic. They’re also usually set up to leverage the enterprise’s identity management system (IAM) for user authentication, which is usually done using the LDAP or RADIUS protocols.

They may also map people into VPN access control groups using basic IAM features (such as group memberships). Some VPNs implement MFA at the moment of connection, and others use host posture checks as extra context to provide dynamic access control.

All of those attributes seem great, and they are all capabilities that a Zero Trust solution must have. So, why are we so pessimistic about workplace VPNs and insist on their replacement?

In the following part, we'll look at VPNs from a Zero Trust perspective, however even from a conventional one, corporate VPNs have a number of flaws.

Although their rules may be configured to manage access to certain IP addresses and ports, this is not the case in most cases. It's significantly easier for network and security teams to assign access to a VLAN or complete subnet, which is likely the same broad access that users have on premises. 2

To be fair, VPNs may be used to provide restricted and targeted access to a small number of corporate resources. This works well for well-defined individuals or user groups that only need access to a limited number of apps. Consider a group of remote workers who are analyzing insurance claims using an internal programme.

These users may only require access to that one programme in order to do their tasks. Consider a third-party vendor who only requires access to a single application. If the programmes they require have static IP addresses, a VPN can be used to offer limited network access in both of these scenarios. Even if this is the case (which is rarely the case for the majority of users), VPNs still have five additional problems.

First, while business IAM can and is used by VPNs for authentication and group membership, access control policies are often relatively rudimentary from an identity standpoint.

For example, regardless of the device from which the user connects, access to a specific set of user credentials will almost always remain the same.

This makes it more difficult for security teams to limit access to personal devices or prevent the misuse of stolen credentials.

Second, from a resource standpoint, VPN access control models are quite static: they're setup to provide access to a single subnet or collection of IP addresses or hostnames.

They aren't built to dynamically resolve target resources and alter user access. Today's IT infrastructures are often dynamic, particularly in firms that use virtualized resources or follow a DevOps strategy. As a result, in order to keep users productive, corporations provide excessively broad network access.

Third, VPNs impose a certain network paradigm on enterprises, as shown in Figure 9-1, by only supporting a single access point onto the company network. This maintains the perimeter-based network architecture, in which all company resources must be linked through an internal network (LAN or WAN). As we've explored throughout the book, this poses a security risk, and achieving it is frequently difficult or impossible in today's dispersed and cloud-based environment. As a result, either the organization's network is left open needlessly, or users are compelled to leave and rejoin to a new VPN server on a regular basis in order to access specified services. End users will be irritated and hampered by the latter.

Fourth, VPN servers must disclose an open port on the Internet in order for users to join. As a result, they are a tempting target for attackers all around the world. Unfortunately, there have been a slew of recent and publicly reported VPN flaws that allow unauthorized remote users to compromise a VPN server and get access to the company network. In today's threat landscape, we believe it is immoral to expose your company network's "*front door*" in this manner.

Finally, VPNs are really just a remote access tool—and hence a silo. They can't be used to enforce on-premises user access controls. For on-premises users, organizations must implement and operate a different set of network and security solutions. This leads in redundant expenditures, effort, and uneven access limits across toolsets (which, in order to avoid limiting user productivity, will most likely result in too-broad network access).

In addition to a usually bad end-user experience, restricted bandwidth, failed connections, and programme conflicts, VPNs plainly have several security flaws. These are the reasons we're so adamant about getting rid of them. Let's compare them to a Zero Trust strategy.

Zero Trust and VPNs

VPNs should be viewed as remote access tools rather than security mechanisms from a Zero Trust perspective. We recognize that this is a divisive position, and that companies can and have had some success with VPNs, but we feel that VPNs have far too many problems to warrant their ongoing use. That is, even a well setup VPN will have constraints that a genuine Zero Trust solution will not. Let's look at how and why this is so. Person access should be dynamically adjusted by a Zero Trust solution based on contextual data about the user, device, network, system, and target resources. The centralized PDP should be in charge of everything. Step-up authentication based on context and user action should also be supported by the solution. The Zero Trust solution should also allow remote identities to access the company network from many locations at the same time. This reduces the need for all dispersed resources to be available from a single point of entry (the traditional perimeter-based security model). As shown in *figure 9.2*, the Zero Trust paradigm allows dispersed PEPs, each covering a logically or physically connected group of resources. The organization does not need to maintain WAN connections between the scattered sites because users may access these PEPs directly:

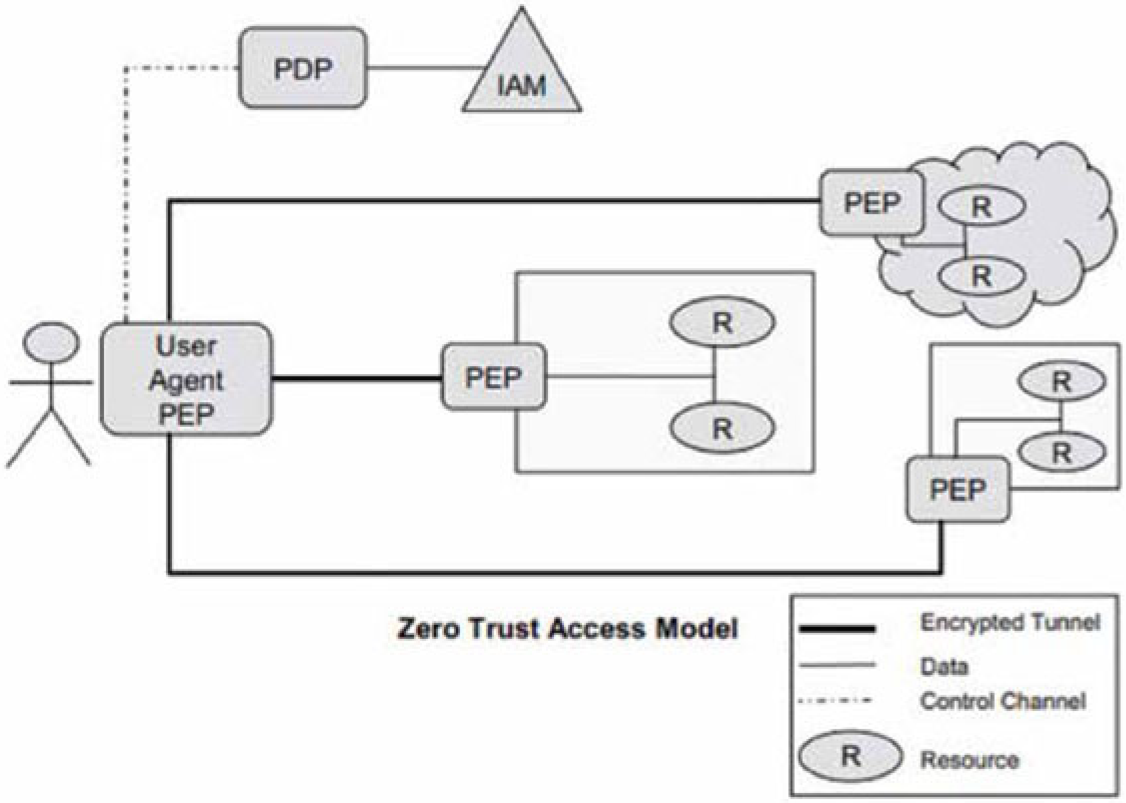


Figure 9.2: *Zero Trust Access Model*

Let us now focus on the two last ways that Zero Trust systems outperform VPNs. For starters, Zero Trust solutions should shield unauthorized users from accessing the company network. Remote entities who do not have authority to access any corporate resources should not be able to view or connect to the network entry point, according to the concept of least privilege. This is a significant step forward in terms of security in and of itself. This may be accomplished in one of two ways: by hiding the network entrance point, as the Software-Defined Perimeter does4, or by transferring the entry point from the enterprise network to a vendor's cloud-hosted platform, as in the cloud-routed approach. Finally, and probably most crucially, Zero Trust uses a single access control paradigm for both on-premises and remote users (by design). VPNs are nothing more than remote access silos, extending the difficulties and inefficiencies that come with being a stand-alone solution. The unified access control paradigm from Zero Trust streamlines operations by providing a centralized platform for defining and enforcing access control policies across all environments.

Before we wrap off this chapter, we'd want to take a quick look at how the various Zero Trust deployment models approach remote access, as they might differ. Both the enclave-based and cloud-routed models enable remote access as part of their designs and will therefore completely replace VPNs. However, remote access capabilities may not be built-in to the other two Zero Trust deployment models, resource-based and microsegmentation. It's critical to have a clear understanding of these needs and any discrepancies when assessing alternative suppliers and architectures, as well as a set of questions to distinguish and evaluate the various options.

Conclusion

VPNs are an antiquated and unsafe method of remote access that must be phased out or replaced when businesses transition to Zero Trust. VPNs are faulty solutions, as we discussed in this chapter, to the point that even well-deployed and well-managed VPN deployments have substantial flaws. It's time for businesses to step up to advance and use a more comprehensive and effective collection of tools to construct their access control models

Your company network and security infrastructure should not have a remote access solution after implementing Zero Trust (enterprise VPN).

It should simply be an access solution that is installed to impose access control for both remote and on-premises users using a single platform and policy architecture. In contrast to VPNs, it celebrates rather than combats the scattered nature of the resources that users access.

CHAPTER 10

Next-Generation Firewalls

This chapter will primarily focus on the role of **Next-Generation Firewalls** (**NGFWs**) in a Zero Trust environment. Most of the key functionalities of NGFW solutions, such as core firewalling, IDS/IPS, and VPN, have previously been described in prior chapters. As a result, rather than a detailed examination of their functionality, this chapter will focus on the role that NGFW capabilities and platforms should play in a Zero Trust future.

Our objective is to assist you understand where and how NGFW solutions should fit into your Zero Trust architecture, as well as how to make sure they're well-integrated with the rest of your company. To accomplish so, we'll start by looking at the market category.

Evolution and history

The traditional 5-tuple firewall rule mentioned in *Chapter 6: Network Infrastructure* was the first enterprise network firewall. It provided a highly narrow set of core network tasks. Traditional firewalls were clearly more focused on networking (allowing or disallowing network packets) with no sense of identification, especially when seen from today's perspective.

Successful firewall suppliers developed over time, and the industry finally agreed on the "Next-Generation" designation.

Almost all business firewalls nowadays are "next generation," with IDS/IPS, traffic analysis, and malware detection for threat detection, URL filtering, and some level of application awareness/control included. Vendors in this market category, like those in the NAC market segment, began their path to identity-centric security about the same time as Zero Trust ideas were circulating in the industry.

segment, began their path to identity-centric security about the same time as Zero Trust ideas were circulating in the industry. Many NGFW vendors now provide Zero Trust capabilities, with varied degrees of effectiveness in satisfying the aforementioned standards. Let's take a look at them from this angle.

Zero Trust and NGFWs

We feel it is fair and appropriate to offer credit to several NGFW suppliers for being early adopters of Zero Trust concepts for on-premises business networks. Their NGFW solutions offer some identity centricity and fine-grained controls, but they fall short of our Zero Trust standards. Most importantly, NGFWs are still firewalls in the sense that they have a restricted range of control. Most importantly, they are not designed to offer security for *all users for all resources regardless of location*. They are primarily hardware-based and do not allow fine-grained remote access, user authentication, encryption, or device isolation (no user agent PEP). Of course, as we've seen in previous chapters, NGFW providers have improved and expanded their platforms through acquisition and organic feature development, adding remote access and other features. While this has undoubtedly been a successful market sector, and there are NGFW vendors with legitimate Zero Trust capabilities, we don't feel it's correct to say the NGFW field has transformed into Zero Trust. Many trustworthy Zero Trust providers didn't start with NGFW in mind, and their systems are built differently.

It's crucial to note that we're not aiming to evaluate individual vendor products or architectures—as we mentioned in the opening, that's a shifting target, and such an assessment would be neither true nor fair. What we're seeking to do is give an explanation and framework so that you can comprehend and evaluate the functional components that generally make up an NGFW from the standpoint of a Zero Trust architecture.

Products classified as NGFWs may or may not be included in a Zero Trust architecture. However, it will very probably incorporate features that have traditionally been part of NGFWs, such as IDS/IPS and a policy enforcement approach that is identity and application aware. As a result, it's critical to discuss NGFWs' position in a Zero Trust architecture. We'll look at two elements of this: first, the ramifications of encrypting network traffic between network components, and second, the overall network architecture that NGFW-based solutions may impose.

Implications of network traffic encryption

Network communication must be encrypted, either inside its native application protocol (for example, HTTPS) or as a result of being routed over an encrypted tunnel, according to Zero Trust principles. While the former is appropriate in some situations (for example, SaaS apps), most Zero Trust solutions rely on an encrypted tunnel into a PEP for a variety of reasons as we described in *Chapter 3: Architectures With Zero Trust*.

The traffic between user agents and network PEPs becomes opaque to any intermediary network component as a result of this. We discussed this problem briefly in our chapter on IDS/IPS, and we'd want to return to it now from a somewhat different angle:

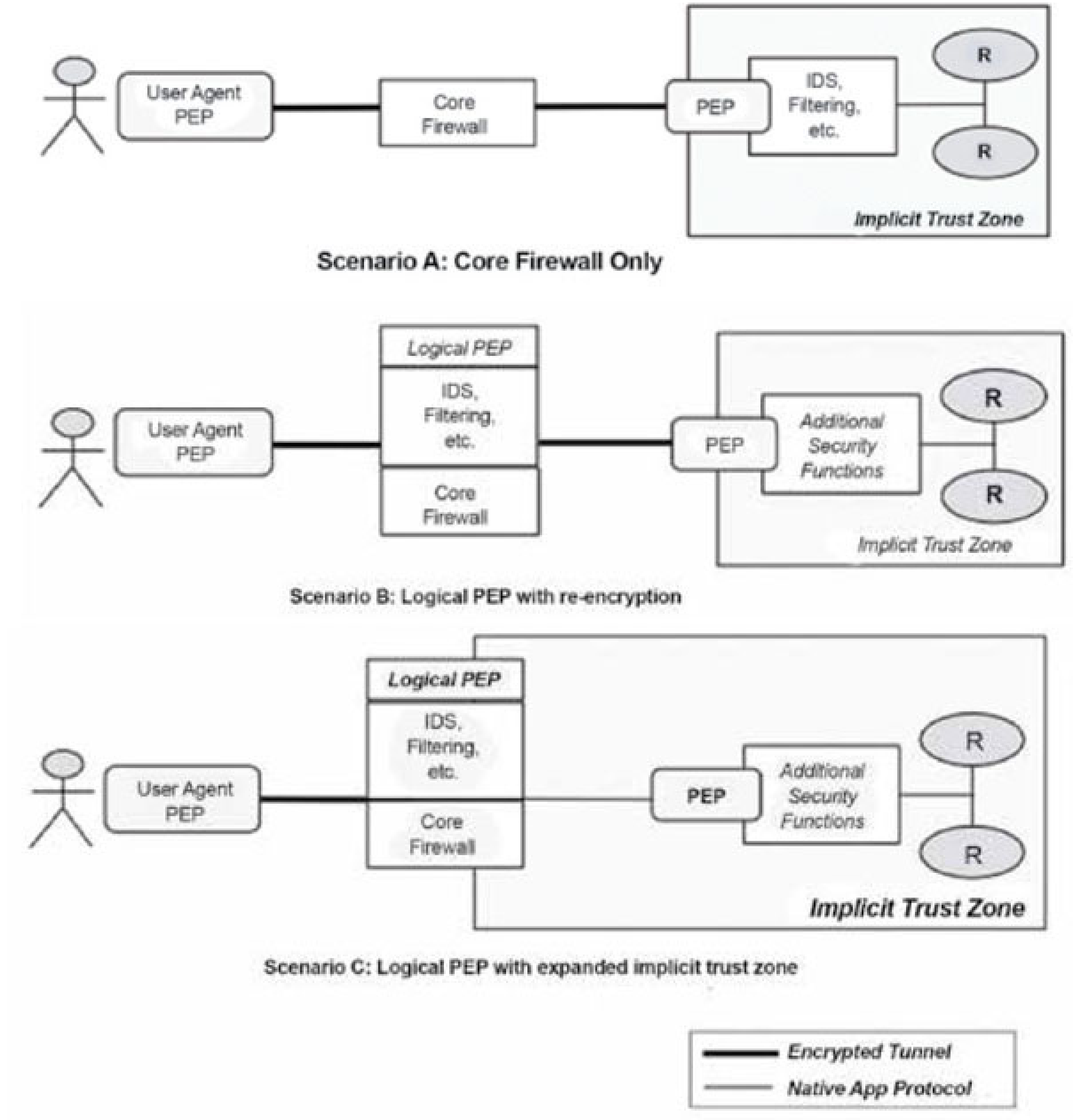


Figure 10.1: *Next-Generation Firewall Deployment Scenarios*

If the networked component is supposed to analyze the payload or take action based on it, it has no choice but to decrypt it, as shown in **Scenario B**. This means that the NGFW is a logical Zero Trust PEP—if it has access to the encryption key, it must be considered part of the Zero Trust platform, in our opinion. This logical PEP provides one or more security functions (such as intrusion detection or URL filtering), which may include application traffic proxying. Scenario B represents the case in which this component re-encrypts network data and sends it over another tunnel to the second PEP and the implicit trust zone. This situation necessitates a significant amount of processing by the NGFW, which would increase network latency and may necessitate a larger (and more costly) appliance to handle the workload.

Alternatively, as indicated in Scenario C, the NGFW can deliver application traffic to the second PEP without re-encrypting it. This lessens the stress on the NGFW (to some extent), but it also results in a larger implicit trust zone, so you should be aware of the ramifications in your environment and network. The current PEP (or components underlying it) can undertake new security enforcement duties in both scenarios B and C. It’s critical to be aware of any potential policy mismatch that the NGFW as a logical PEP and the second PEP are enforcing. This is especially critical if the security components come from various manufacturers and have distinct policy models. None of this is meant to indicate that NGFW providers’ Zero Trust systems are necessarily better or more successful than alternatives; in fact, as we’ll see next, there are additional trade-offs and factors to consider.

Architectures of networks

It’s vital in any Zero Trust design that your team understands the entire network topology of any solution and how it fits into your company network architecture. In some aspects, such infrastructures are constantly evolving—for example, through the use of cloud-based resources—and in other others, they are quite static or unchanging, such as WAN lines that have been in place for years.

We’re looking into this since some NGFW-based solutions may demand specific network designs or impose certain limits, limiting your ability to progress easily on your Zero Trust path. Take a look at *figure 10.2* for two instances of Zero Trust network designs:

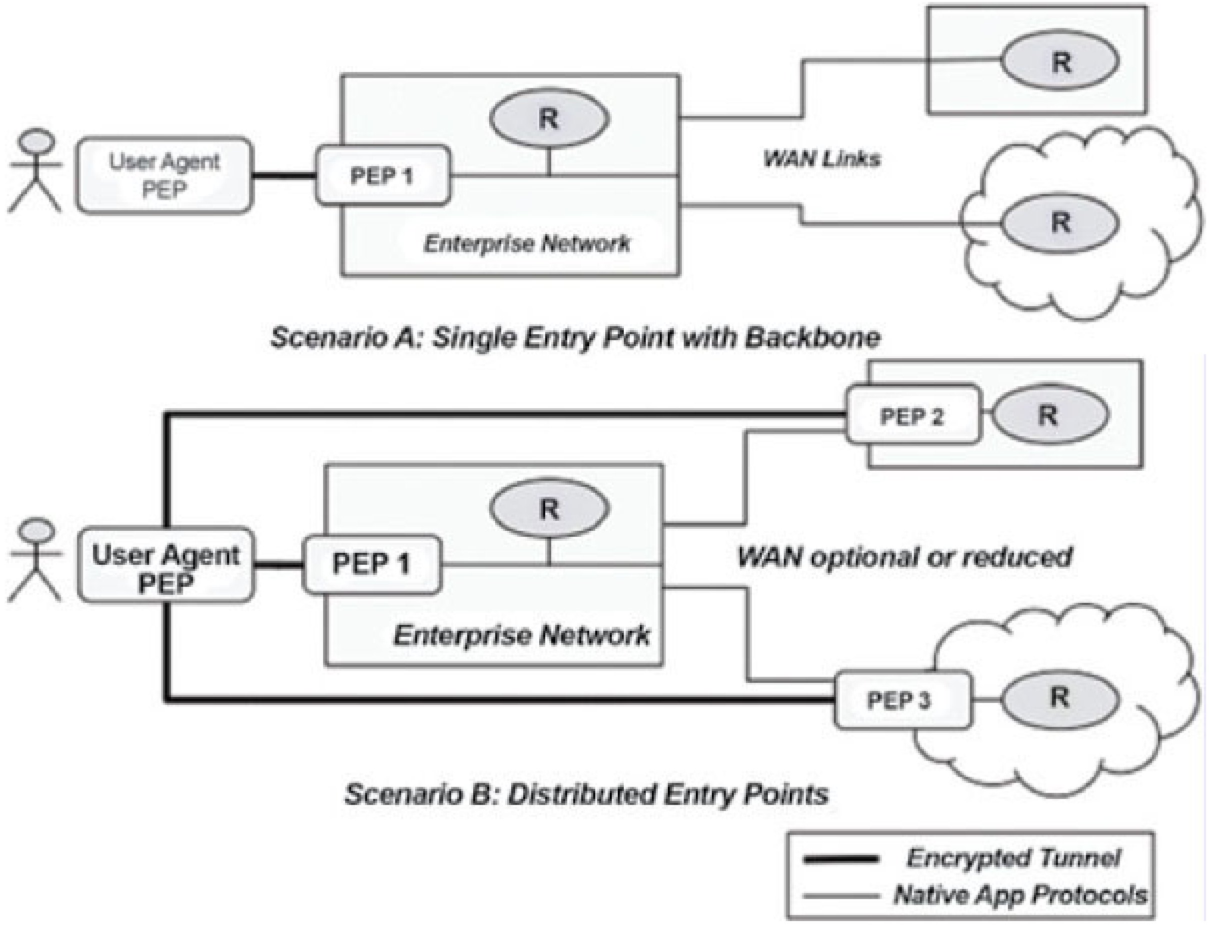


Figure 10.2: *Zero Trust Network Architectures*

Scenario A depicts an architecture enforced by NGFW-based Zero Trust systems, with a single access point for distant users into the company network.

A **Wide Area Network** (**WAN**) or backbone is required for distributed resources (which are used by all modern businesses nowadays). Simple firewalls will be installed at the distant networks' ingress points, imposing basic network **Access Control Lists** (**ACL**).

The problem with this strategy is that it reinforces the idea of a hard border with a soft inside network. This architecture falls short of our objectives since PEP1 is the only place where Zero Trust tenets are applied. There are two more concerns with this strategy.

First, the WAN adds to network delay by basically forcing all user traffic to be backhauled to the PEP 1 entry point. Of course, WAN bandwidth comes at a cost to the company. Second, this technique may result in a loss of policy fidelity—PEP1 is so “*far away*” from the resources in remote places that it is impossible for it to enforce fine-grained or dynamic access regulations effectively.

Consider Scenario B, which uses scattered entry points and requires users to connect directly to their authorized PEPs. This eliminates the requirement for user traffic to be backhauled to PEP1, lowering latency and WAN expenses. Organizations may drastically cut their WAN consumption, and in certain cases, altogether remove it by replacing it with basic Internet access.

This has the added benefit of maintaining perfect integrity, as all PEPs have complete control over fine-grained, identity-centric, dynamic rules applied to their local resources. PEP 2 and PEP 3 may also perform API requests and learn characteristics about their protected environments and resources since they are complete Zero Trust enforcement nodes.

Keep in mind that real architectures may differ from this—many manufacturers may provide a blended or hybrid approach, and your company will undoubtedly have its own unique characteristics. We recommend that you ask smart questions and devote the time and effort necessary to fully comprehend your present and projected network topology from the viewpoints mentioned below.

Conclusion

To summarize, we anticipate that the adoption of Zero Trust will have a substantial influence on the NGFW industry, blurring the borders between what was formerly a distinct and well-defined category. We’re seeing NGFW providers add Zero Trust capabilities, just as the *business firewall* industry developed to the point where almost every corporate firewall now contains features that were formerly called **next-gen**.

Enterprises will increasingly implement network security solutions that incorporate Zero Trust concepts in the future, which by definition necessitate a broader perspective and policy model. This is a good thing, because enterprises are increasingly wanting to install fewer solutions that cover a larger region, as they know that fragmented security solutions are incompatible with a Zero Trust strategy. As a result, they must make certain. As a result, they should make sure that the security components they choose provide comprehensive APIs and can be readily integrated (one of our extended Zero Trust principles). The sources of identity and context available to the PDP, as well as how broadly the policy model may be implemented via PEPs spread across business assets, are the major choices in our thoughts when it comes to a Zero Trust architecture. There is currently no commercially available platform that has a policy model and set of PEPs that can be implemented across all users, infrastructure, and use cases. One of the reasons Zero Trust is a journey is because of this. This emphasizes the significance of making informed decisions, ensuring that your platforms and tools are well-suited to your first use cases and can be linked with the rest of your environment and PEPs. It's possible that you'll employ an NGFW vendor's platform as the foundation of your Zero Trust architecture, and that's a good idea. Simply be aware of the platform's limitations (boundaries), ask some tough questions about how it can fit into your larger ecosystem, and be mindful of any architectural constraints. You'll have PEPs that aren't part of the vendor's platform but must be incorporated. Make sure your selected platform can support this in your environment efficiently and effectively.

Chapter 11

Security Operations

Many businesses have invested in establishing a **Security Operations Center** (**SOC**) as a real or virtual organization to manage threats, vulnerabilities, and incident response. **Security Information and Event Management** (**SIEM**) and **Security Orchestration, Automation, and Response** (**SOAR**) are the two main tools used in SOCs in this chapter. We'll look at these tools through the lens of Zero Trust, seeing how they may work together to increase the efficacy and efficiency of day-to-day operations in the SOC. But before we can connect these systems, let's go through why SIEM and SOAR exist in the first place.

Modern IT systems produce massive amounts of log data in a variety of forms, locations, and schemas. These logs are used for a variety of purposes, including periodic IT access for troubleshooting or diagnostics, ongoing anomaly detection, and long-term archives for forensic or auditing. These logs not only give a comprehensive picture of infrastructure components and their interactions, but they also enable SOC analysts and tools to evaluate and synthesize events throughout the whole IT system. SIEM solutions have evolved to handle massive quantities and a wide range of log data, and they have become an essential component of current SOCs.

Security analysts, of course, do a lot more than look at logs; they also spend a lot of time and effort on incident response and event management.

Fortunately, SOAR tools offer automated (or at least semi-automated) procedures that may be swiftly integrated to satisfy incident response needs throughout the SOC's whole tool set. Because SOC operations are so important to an organization's security programme, the rising capabilities of SOAR technologies will help SOC teams become more effective by transforming how they use the massive volume of data and number of security incidents.

SIEMs and SOARs are progressively becoming integral aspects of a SOC in practice.

As identity context becomes more prevalent throughout the security ecosystem, the value of SIEMs and SOARs will only increase. In this chapter, we explain how and why these tools benefit from being well-integrated into a Zero Trust architecture.

Security Information and Event Management (SIEM)

SIEM technologies allow for the collection, aggregation, and normalization of log data in order to identify and evaluate security incidents inside an organization. For decades, IT firms have used logs and aggregated log management systems, and the security-specific market segment, today known as SIEM, arose about 2005. SIEM companies created a set of security-focused features for unifying, normalizing, aggregating, correlating, and analyzing log data, converting it into security information and (hopefully) actionable events, going beyond simple log management.

This log data is often ingested from security systems such as IDS/IPS, endpoint management, authentication systems, and others, in addition to IT infrastructure (servers, firewalls, and so on). Large amounts of log data are generated by these business systems and networks, which may be daunting for analysts. SIEMs can assist sort this out by providing analytics, filters, visualizations, and other tools, such as filters, visualizations, and other tools that can help decrease the number of false positives.

SIEM providers have traditionally been implemented on-premises but have lately switched to a cloud-based paradigm. The two SIEM deployment types (on-prem vs. cloud) have advantages and disadvantages, but from a Zero Trust viewpoint, these distinctions are mostly irrelevant—the integration scenarios, needs, and benefits we cover later are similar. That is, regardless of where your SIEM is located, incorporating it into your Zero Trust architecture may provide significant benefit.

SIEMs can help map an organization’s network infrastructure by synthesizing raw data in addition to aggregating logs. This can help security and IT teams by giving them a better understanding of where events are occurring in a network. This is intriguing because it begins to provide information about the organization's high-value (or at least, highly utilized) assets, which can help organizations better define and plan their Zero Trust strategy and architecture, for example, by influencing policy definitions or PEP deployment locations. SIEMs by themselves have proven extremely beneficial, enabling data aggregation and \saiding in decision-making for security analysts, and a logical extension for these \splatforms has been to give structured and event-driven techniques to automate reactions \sand actions to identified events. These skills have come together to form SOAR, a collection of services.

Security Orchestration, Automation, and Response (SONAR)

SOARs are frequently used in conjunction with SIEMs, and they're occasionally offered by the same vendor as part of a unified platform. A SOAR will ingest data from a SIEM (such as observed events or threshold alerts) and offer a model and method for automating a sequence of reaction activities, generally led by machine learning.

This is beneficial because SOARs give a common context for events as they filter through the enormous number of events broadcast by a SIEM, and eventually automate procedures or workflow in response to the event.

This integrated automation reduces the number of false positives in an environment, allowing security engineers to focus on actual issues.

The value of SOAR is not only in the automation, but also in the modelling of logical analysis and reaction channels. These workflows contain information about an organization's networks, systems, and dependencies, as well as how to interact with them—information that is all too often "tribal knowledge" held primarily in the minds of senior analysts. With a SOAR, this information can be turned into an automated, repeatable, and dependable platform that doesn't require any downtime. A SOC may (and should) be elevated to a seamless integration of people, process, and technology thanks to this codified knowledge.

Achieving these principles from a Zero Trust perspective necessitates more than standalone technologies—it necessitates integration and coordination, as well as "*reach*" to affect changes across the enterprise security infrastructure—something that a SOAR is well suited to achieve when connected to a Zero Trust platform. SOARs, in particular, assist SOCs in achieving their purpose by automating repetitive, predictable procedures. While actively acquiring threat information, reacting to data, and providing context to data, most SOARs will detect decision patterns and assist in managing the whole incident response lifecycle. Vulnerability management2 and threat intelligence are also important roles in a SOC, with the SOAR offering a good workflow and incident response pattern to enable these, and the results contributing to the SOAR solution's ongoing growth and learning. SIEM and SOAR's analysis and actions are critical components of a good Zero Trust system, serving as additional context and accelerators for PDP choices, which we'll go over in more detail in the following section.

Zero Trust in the Security Operations Center

SIEMs and SOARs will continue to be important components of corporate security, and their value and importance will grow as enterprises embrace Zero Trust. That is, as businesses get closer to Zero Trust, they should expect (and demand) increased SIEM/SOAR breadth, depth, and overall efficacy. Furthermore, the automatic learning provided by a Zero Trust-integrated SOAR will only help the PDP make better judgments in order to support the whole environment. Let's have a look at how this happens.

Enriched log data

SIEMs' capacity to correlate data from disparate systems, such as the assignment of a dynamic IP address to a user and subsequent network activity done by that IP address, is one of its most important functions. SIEMs, on the other hand, are restricted to the data given by their source systems and are frequently hampered by underlying technological restrictions and segregated infrastructure parts. Network access, for example, frequently crosses network boundaries where **Network Address**

Translation (**NAT**) happens, making it difficult or impossible to attribute IP address activities to a single user.

Furthermore, logs are frequently created by systems that use silos or unconnected identity management systems, making it difficult for SIEMs and security analysts to combine or disambiguate user IDs across several log sources. Zero Trust systems not only remove many of these technological limits, but they also vastly enhance the amount of data absorbed by SIEMs, enhancing their capacity to correlate and identify security-related events. A Zero Trust system, in particular, will be able to record precise identity enriched data into a SIEM since it is primarily identity centric. This improved log data will be more useful to SIEM and SOC engineers, allowing them to respond more quickly. In other words, a Zero Trust system should be able to log all network access by all users and augment that data with information about their identities, devices, and general context. This should be true regardless of where a user is situated, how many intermediary network boundaries are traversed, what network protocol is used, or how the identification system of a given application refers to a specific user.

Orchestration and automation (triggers and events)

Enterprise Zero Trust solutions must be highly automated, with the capacity to identify and respond to a variety of triggers and events at a large scale. The dynamic feature of the Zero Trust policy model, as we've highlighted throughout, is a big contribution to its value. SOAR systems can augment and increase the efficacy of a Zero Trust system since they have a larger reach than Zero Trust. Combining SOAR with Zero Trust via a series of synchronized events, orchestrated API requests, and triggers will really improve both systems. The specifics of which component performs which role may vary depending on your architecture and platforms, but processes coordinated between a PDP and a SOC security analyst will inform and execute real-time choices and actions. In the next sections, we'll look at a few instances. Of course, to conduct operations like transferring updated data, initiating a policy assessment refresh, or programmatically generating new policies or virtual infrastructure components, this connection requires a set of bidirectional APIs. We'll go over things in greater detail in Chapter 17, but we wanted to give you an outline now because they're important to this conversation. There are four basic types of triggers in a Zero Trust system, which are logical methods to communicate with external systems such as SIEMS and SOARs. The Zero Trust system initiates three of them, while an external system initiates the fourth.

Authentication trigger

This usually happens just once or twice each day for users. This is far less common for services (non-person entities). This trigger of course triggers a policy assessment by the PDP and is a natural moment for the PDP to send queries to a SIEM/SOAR to collect further user or environmental context.

Resource access trigger

Identities, of course, use a PEP to access resources several times during the day. It's common for a PEP to make periodic calls to a SIEM/SOAR to get up-to-date context, particularly for properties that may have changed after authentication, such as the device risk rating based on observed behavior. PEPs shouldn't have to reevaluate everything every time they log in, so think about how your Zero Trust system models this trigger.

Periodic (session expiration) trigger

Many Zero Trust systems have an identity session idea, which has a finite duration (such as several hours). When a session expires, Zero Trust systems frequently renew the identity's assigned policies, and this is a natural moment for the PDP to make calls to the SIEM/SOAR to gather further context, similar to when the session is authenticated.

External trigger

Finally, many Zero Trust systems provide inbound APIs that may be used by other components to initiate events and update contextual data. In order to get the most out of a Zero Trust system, your SIEM/SOAR must provide a corresponding set of both inbound and outgoing APIs. When comparing Zero Trust systems, look for one that has a large number of actions to support a wide range of integrations. To put this into context, let's look at three integrations between Zero Trust and SIEM/SOAR.

Zero Trust querying for additional context (authentication trigger)

In our first example, a Zero Trust system makes API calls into a SIEM during user authentication, as shown in *figure 11.1*. This integration is intended to provide the PDP more information to help it make better decisions:

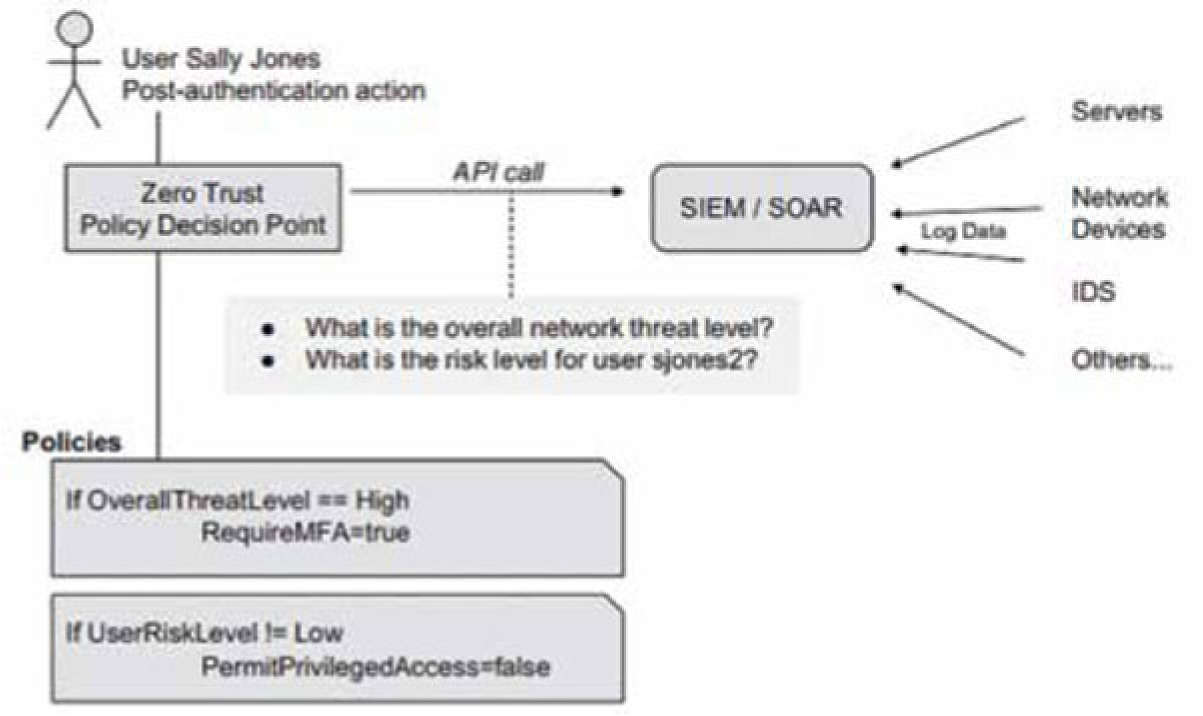

User Sally Jones
Post-authentication action
Zero Trust
Policy Decision Point
API call
SIEM / SOAR
Log Data
Servers
Network
Devices
IDS
Others...
• What is the overall network threat level?
• What is the risk level for user sjones2?
Policies
If OverallThreatLevel == High
RequireMFA=true
If UserRiskLevel != Low
PermitPrivilegedAccess=false

Figure 11.1: *Zero Trust System Making Decision Based on SIEM/SOAR*

In this case, when Sally has successfully authenticated, the Zero Trust system is acting as a PDP, reviewing policies, and making choices about which resources Sally should be allowed to access at this moment, depending on relevant contextual information. The Zero Trust system in our example uses two characteristics obtained through API call from the SIEM system: the general threat level on the network and the risk level connected with Sally.

The policies presented analyze these characteristics and apply them to the Zero Trust system's enforcement mechanisms. If the SIEM indicates that the overall network danger level is High, the first policy needs MFA.

The second policy bans users from accessing resources that need privileged access if they have been marked as not presently having a low risk level—for example, based on device posture or observed network activity.

In response to the authentication trigger, the PDP queries the SIEM/SOAR, as seen in the preceding example. The ability to run a similar query based on the session expiry trigger, as well as the resource access trigger, will enhance the Zero Trust system. Let's take a look at an API request that goes the other way.

SIEM/SOAR invoking Zero Trust System (external trigger)

This example demonstrates how a SOAR system makes an API request to the PDP to start a process. This action is initiated when the SOAR system does some analysis and determines that something is wrong with a server, a user's device, or the network, and that action is required as a result.

This call, as illustrated in *figure 11.2*, may contain information that is particular to a single user or information that is relevant more generally (such as the overall threat level for the network):

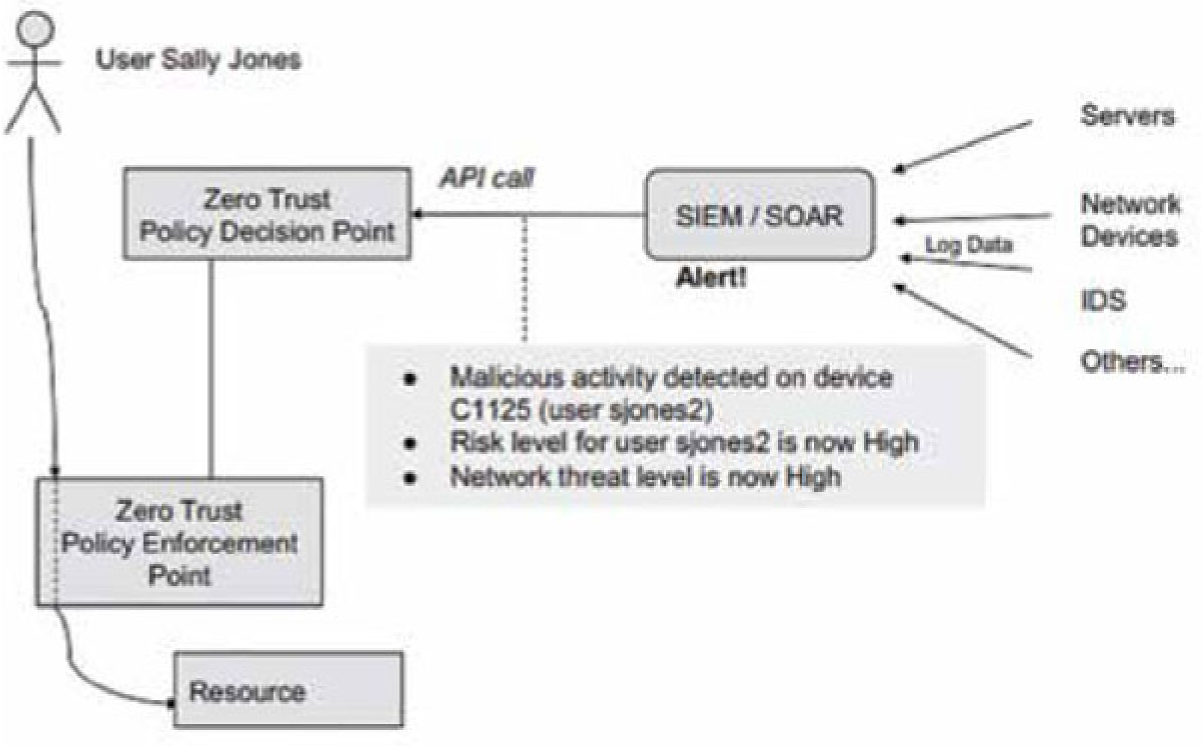

User Sally Jones
Zero Trust Policy Decision Point
API call
SIEM / SOAR
Alert!
Log Data
Servers
Network Devices
IDS
Others...
Malicious activity detected on device C1125 (user sjones2)
Risk level for user sjones2 is now High
Network threat level is now High
Zero Trust Policy Enforcement Point
Resource

Figure 11.2: *SOAR Initiating Workflow/Response*

Of course, the Zero Trust system must be able to reply correctly to the SOAR's API call depending on rules in this circumstance. For example, if the Zero Trust system notices unusual activity on Sally's device, it may prompt her to do one of the following:

- Re-authenticate on that device
- Request MFA from Sally
- Restricting Sally's device access as soon as possible, such as by quarantining it on the network

Indirect integration (external trigger)

Finally, a word on how the SIEM and the Zero Trust system interact. While the above example appears easy, it is really rather complicated behind the scenes, since it necessitates configuring the SIEM/SOAR to know which data the Zero Trust need in order to analyze its policies. Because there is now a bidirectional data reliance between these two systems, this adds complexity and operational overhead to the system. If a policy in the Zero Trust system is set up to use a new attribute from the SIEM/SOAR, the

SIEM/SOAR must be updated to include that attribute in its API calls to the Zero Trust system as well. This necessitates coordinated modifications on both sides, which increases the level of difficulty. An alternate, and more straightforward, option is for the Zero Trust system to request the data it requires from the SIEM/SOAR. Modifications to a Zero Trust policy won't necessitate any changes to the SIEM—as long as it has the data requested, it'll be able to supply it. The *figure 11.3* illustrates this concept:

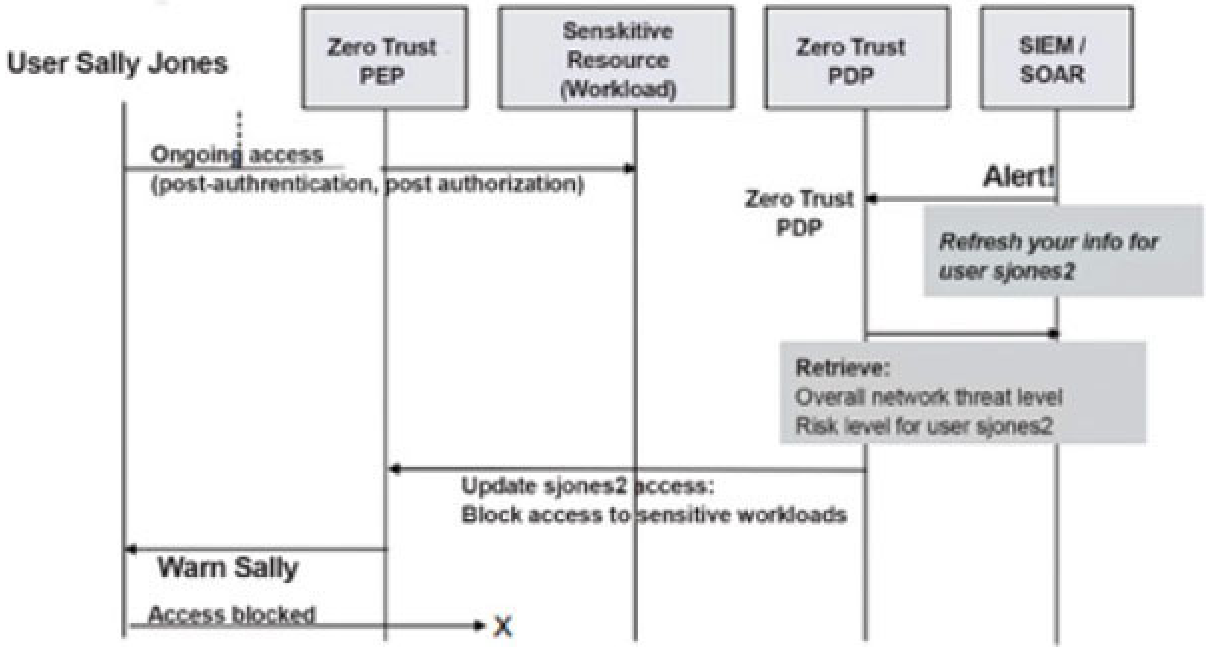

User Sally Jones
Zero Trust PEP
Senskitive Resource (Workload)
Zero Trust PDP
SIEM / SOAR
Ongoing access (post-authrentication, post authorization)
Alert!
Zero Trust PDP
Refresh your info for user sjones2
Retrieve:
Overall network threat level
Risk level for user sjones2
Update sjones2 access:
Block access to sensitive workloads
Warn Sally
Access blocked
X

Figure 11.3: *Swimlane of SOAR and PEP Interaction*

Sally has already authenticated and been permitted to access a sensitive task in this diagram, which is portrayed as a swim lane for clarity. The SOAR system then detects unusual behavior linked with Sally or her device and sends a simple API request to the Zero Trust PDP, informing it that something has changed and that the Zero Trust system has to update Sally's information (user sjones2). The Zero Trust system then reacts based on that API request, most likely by re-evaluating Sally's whole set of rules, which involves getting new information about her from numerous systems, including the SOAR. It's worth noting that the Zero Trust system, not the SOAR, determines which data pieces the PDP need to analyze its policies. This eliminates the requirement for the SOAR to be aware of what data elements the PDP requires to review its policies.

The PDP decides that Sally should no longer have access to the sensitive resource based on this new knowledge, and it tells the PEP of this decision. The security team has also opted to alert Sally, possibly via a pop-up message or an SMS in our case.

Conclusion

SIEM and SOAR solutions have become essential components of modern SOCs, providing security analysts with crucial analysis, visualization, and reaction capabilities. The SIEM or SOAR can (and should) play a key role in bringing solutions together for quick and near-real-time analysis and reaction in a Zero Trust architecture.

The integration scenarios we described here show how these systems may be brought together at various times and based on various triggers to increase security and response efficiency and effectiveness. These are only a few examples; there are many more ways that Zero Trust systems and SIEM/SOAR may work together to do useful and interesting tasks.

Examine how your SOC team uses the tools they have and teach them about the kind of identity and context-enriched data your Zero Trust system can provide to their platforms. It’s highly probable that the two of you will come up with a slew of ways in which these integrations might benefit your company. And \shaving the SOC team on board can only help expedite your Zero Trust journey.

Chapter 12

Privileged Access Management (PAM)

Privileged Access Management (PAM) is a segment of the IT security business that uses a collection of security functions and procedures to govern, manage, and report on how privileged users (system administrators) access systems and resources. PAM may be used to control access to any system, although it's most commonly employed to protect high-value assets like domain controllers and production servers. Of course, Zero Trust security is built on the idea that it should be used to secure all systems, but high-value systems—which are often already protected by PAM—are suitable candidates for early Zero Trust projects or scope.

As the industry has grown, PAM solutions have evolved and extended to include capabilities such as password vaulting, secret sharing, and session management. While PAM may frequently employ group membership to manage access and authenticates users with business identity providers, we feel it is more appropriate to classify these solutions as identity-aware rather than identity-centric.

This distinction is important because, while a PAM solution may resemble a Zero Trust system in certain aspects and perhaps include some PEP-like features, it cannot be regarded a Zero Trust solution on its own. We'll come back to this issue at the end of the chapter, but first, let's look at the three fundamental services that PAM normally provides.

Password vaulting

PAM solutions originated with the simple notion of a password vault—rather than depending on admin users to keep track of privileged account passwords individually, these credentials are saved in a safe repository. Passwords are securely stored and managed in the vault, which also automates their lifetime, including expiration and rotation. These systems implement the necessary business processes, such as access request and approval processes for "*checking out*" passwords for use (in the past, password vaults functioned as a password library, with users "checking out" passwords in the same way that patrons check out books from a library). In today's world, credentials are frequently ephemeral, and they are routinely cycled once the specified term has passed.

Users may never see their password since they are automatically authenticated by the PAM system, which signs them into the target system behind the scenes. Password vaulting has progressed beyond just storing privileged account credentials to delivering passwords via APIs to support service accounts, as well as password management for these accounts. These API capabilities allow programmes, scripts, and service accounts to avoid storing passwords in plain text or in risky areas.

Password vaulting is useful in the context of PAM since it allows you to accomplish many purposes. First, it establishes a least-privileged access paradigm for credentials, which is an evident feature of Zero Trust contexts.

Second, it aids in the enforcement of business processes for getting sensitive resource access. Finally, it assures that privileged system access is documented and auditable, which is critical in many regulated contexts.

Secrets management

PAM systems have evolved over time to offer a larger range of secret management capabilities, moving away from storing and managing relatively basic user passwords. Secrets do not have to be passwords; they may be any sort of information required to safeguard systems directly or indirectly. Other than passwords, the following are examples of items that might be saved in a secret sharing solution:

- Hashes
- Certificates
- Cloud tenant information
- API keys
- Database storage information

- Personal Information
- SSH connection information

What they all have in common is the necessity to securely store sensitive data in a way that only authenticated and authorized users may access it while maintaining data integrity (that is, cannot be tampered with). Secrets management solutions must provide safe and auditable access for both users and systems.

There are also business and process-oriented benefits to secrets management in addition to the technological ones—specifically, the workflows and procedures put in place to both store and access these secrets. Because there is a (controlled) location and safe storage, companies may avoid storing credentials haphazardly, lowering the chance of them being stolen or lost.

Finally, as previously noted, non-person entities can utilize API techniques to access the secrets management location to automate the retrieval of secrets in an application or server during the bootstrapping of that environment.

Privileged Session Management (PSM)

Privileged Session Management (PSM) is one of the most significant parts of PAM, especially because it isn't usually included as part of a Zero Trust solution. Rather than a rigorous security driver, compliance needs, and audit difficulties are frequently the reasons that compel firms to pay for and execute a PSM solution. PSM solutions effectively intercept or proxy system administrator access to target systems, allowing administrators to be monitored, recorded, and restricted using protocols like **Remote Desktop Protocol** (**RDP**) and **Secure Shell** (**SSH**).

For businesses, PSM solutions generally serve two major purposes. For starters, they can give admin access keylogging or session recording, guaranteeing that all such operations are captured for audit, compliance, and forensic purposes.

Second, they can give "*supervised*" admin access, in which a second person can monitor an admin's session in real time to ensure that high-risk operations are not overlooked.

PSM is also frequently used to enforce role-based access on a privileged system, by granting users just the rights they need to complete their duties. This can take the shape of limiting the admin's account's rights or outright preventing specific commands from being executed on the target device. Consider the situation when a Windows developer has to deploy code and subsequently restart a specific site on an IIS server but is not allowed to use the IISRESET command. The PSM can guarantee that the position they've been given just has the bare minimum of permissions.

Another example for a Linux system is having the session management system block users from using the ssh command to migrate laterally.

The *figure 12.1* depicts one method a PAM system may be built, with a central PAM policy server and a distributed collection of PAM agents running on production servers, to summarize our introduction to PAM (the protected resources). In this case, the business could have opted to employ their PAM solution instead of other options like jump boxes:

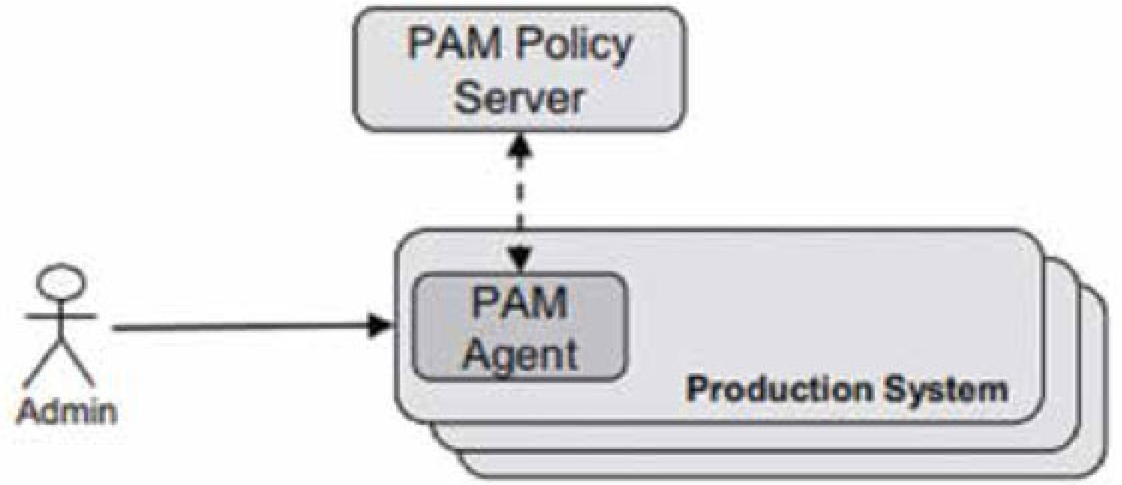


Figure 12.1: *PAM Providing Access Control Through Session Management*

The agent gets information from the policy server in this example, which describes what rights a certain user has on the target system. While the user has direct access to the server, the agent controls who may log in and may also give RBAC and control over administrative operations. It's worth noting that we're depicting the PAM components using terminology that's consistent with Zero Trust because they share certain similarities. There are a few key changes as well, which we'll go over in the following part. Finally, in the future (and beyond Zero Trust), the rising use of serverless computing and DevOps-style "*immutable infrastructure*" is altering the methods in which administrators undertake privileged activities, making conventional PSM (and, to a lesser extent, password vaulting) less important.

As companies adopt this mindset, they adapt such that administrators never have to log into a production system to do a manual activity. When done correctly, this leads to speedier and more reliable outputs, since more is driven "*as code*" and less is done manually. Note that we'll return to this topic in *Chapter 18*: *Zero Trust Scenarios* of the DevOps scenario.

Zero Trust and PAM

Let's talk about how the pieces of PAM exist within a Zero Trust environment now that we've looked at PAM from the standpoint of traditional IT and security. Keep in mind that, while PAM features (vaulting, secrets, and session recording) will continue to play an essential part in security designs, within a Zero Trust environment, there may be some modifications (and probable diminishment).

Many PAM systems already include a built-in policy and access model and can interface with identity providers for user authentication, role-based access control, and attribute-based access control, as we described earlier. In this approach, they serve as policy enforcement points to some extent.

But first, let's tackle PAM's "800-pound gorilla": password vaulting. Because of the non-Zero Trust approach of a too-open network, where every user has ongoing network access to every server, a vault with server password obfuscation and rotation is necessary. With Zero Trust, this premise is no longer valid! In principle, in a Zero Trust network, you might eliminate passwords for privileged server access and instead depend on PEPs to enforce Zero Trust regulations based on context and business processes. Now, we're not proposing you do this, but it's an interesting point to consider, and it demonstrates how a Zero Trust network might change the value proposition of a password vault. Although we do not propose deliberately decommissioning PAM vaults, you should avoid utilizing them for new settings and projects. Keep in mind that the other features of PAM, including as secret management and session recording, will continue to be useful in a Zero Trust environment.

Let's look at how PAM connects to Zero Trust in more detail. Protecting access to the PAM server itself by placing it behind a PEP, as shown in figure 12.2, is the most easy and easiest way. The PAM solution is a protected resource within the Zero Trust architecture in this instance. Increasing the security of the PAM solution by blocking unauthorized people or devices from accessing it, while simple, is nonetheless reasonable and beneficial. After all, if the PAM server has the *keys to the kingdom*, it's an obvious target for malevolent actors:

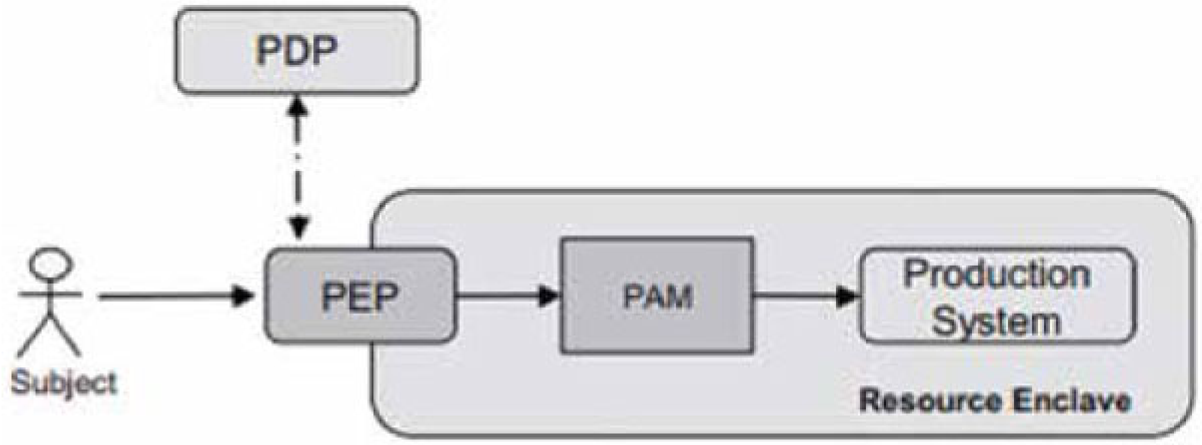


Figure 12.2: *Deploying the PAM Behind a PEP*

Moving beyond this simple case, consider how PAM may work better with a Zero Trust solution, for as by using identity context or assisting with policy enforcement:

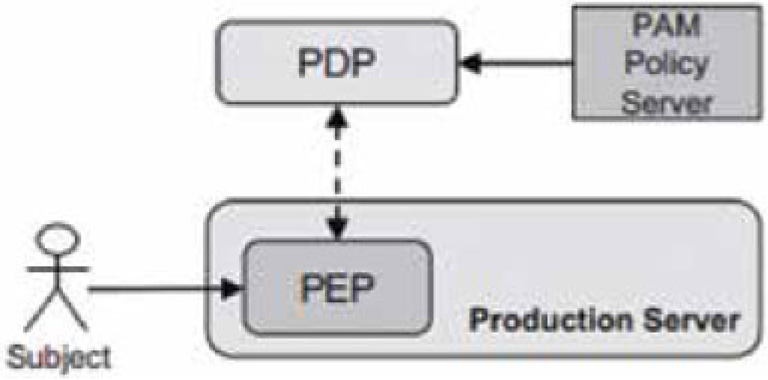


Figure 12.3: *PAM Integration with Zero Trust*

The *figure 12.3* depicts one such integration, illustrating how a PDP might be integrated with and consume PAM information or policies for use in the Zero Trust policy paradigm. This integration might be as easy as telling the PDP which high-value servers require tighter authentication or device posture checks using the PAM. It might also be a more complex integration, in which the PDP consumes PAM-defined policies on which administrators should be allowed to access certain servers and passes them on to the PEP for enforcement.

The *figure 12.4* demonstrates another situation in which the PAM consumes data from the PDP and uses it to assist better enforce access rules. This might include identification or device attributes that can help you decide whether or not to provide access to the target system. Most PAM systems, for example, lack remote access capabilities, but Zero Trust options have. A PDP may provide the PAM with the user's geolocation information, which the PAM could use to determine whether or not to provide access:

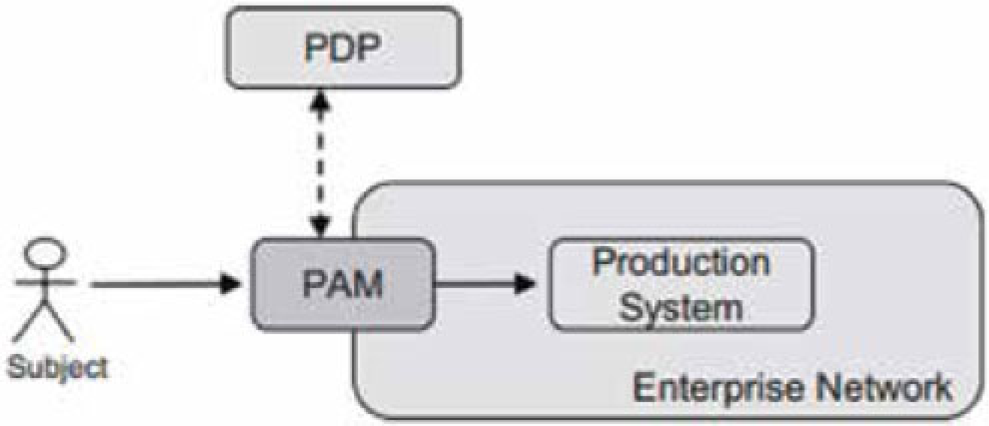


Figure 12.4: *PAM Consuming Zero Trust Context from PDP*

While these final two situations are more hypothetical than real-world scenarios, we expect that as Zero Trust systems become more widely used, they will become more open, allowing for these kinds of interfaces across different security components from different suppliers. If PAM companies grow into the Zero Trust space, they may appear more quickly.

This, we believe, further emphasizes how today's PAM systems are more identity-aware, rather than identity-centric. While they frequently employ an enterprise identity provider to authenticate users and can establish access restrictions based on group membership, this is usually the extent of their capabilities.

The *figures 12.3* and *figure 12.4* depict forward-looking scenarios that will aid in bringing PAM solutions into the dynamic and identity-centric world of Zero Trust.

Conclusion

PAM obviously delivers essential password and access management services, as well as the ability to assist with security, compliance, and audit requirements. It can assist accomplish certain parts of the concept of least privilege and can employ identity attributes to help manage access, but it’s not a complete Zero Trust platform. However, combining a PAM with a Zero Trust platform may boost the value of both systems, and firms using a PAM should prioritise safeguarding the PAM server. They may also consider how these two systems could share information in order to make better access decisions jointly, however this is more likely to be a more advanced or future use case.

Chapter 13

Data Protection

Forrester's **Zero Trust eXtended** (**ZTX**) methodology puts data at the core, and for good reason: important data exists in every business, and it must be secured. Data, which is frequently the primary target of attackers, is a critical organizational resource that PEPs must safeguard through integration with a PDP, using an identity and metadata-centric policy approach.

High-value data is now routinely stored, accessed, and processed across a number of platforms, including on-premises, cloud-based, and mobile devices, as data quantities have risen dramatically in most enterprises. As businesses move to the cloud and undergo digital revolutions, the volume and complexity of data will only increase.

Through appropriate data lifecycle and usage activities, this expansion must be successfully controlled and safeguarded. We'll talk about the data lifecycle, data protection, and data consumption (including labelling and categorization) in this chapter, and how data security fits into a Zero Trust approach.

Data types and data classification

Structured and unstructured data are the two categories of data that most people are familiar with. The distinction is critical because it influences how security may be implemented and how technology can be utilized to assist data security.

Structured data is information that is kept in a database and can be retrieved and generated using a defined method (for example, via Structured Query Language, SQL).

The actual method of storing data in a database is controlled by the database technology used, although it's usually done in binary format with access control established inside the database system. Databases, on the other hand, often follow a predetermined structure that limits the sorts of data that may be stored and assigns information such as column names.

A database table containing employee records, for example, might have a number of columns defined, such as date of birth (date type), street address (free-form text), and employee ID (integer type). This imposes an implicit degree of categorization associated with the table's columns of data, which gives information on the table's security needs and how it should be handled.

Unstructured data is data that is produced in an ad hoc manner and prepared by the user or the data storage technology. Most significantly, unstructured data does not fit into any preset schema, making it impossible to specify security criteria and categorization automatically, either globally or per-field.

That is, because the files are unstructured, they do not convey information about the material contained inside them by default. A document does not expressly state that it contains employee birth date information, unlike the date of birth database field example. Furthermore, unstructured files differ from database data in terms of security. Files on a file share may not be encrypted or obfuscated in any way other than by the software that produced them. While the data may be in a proprietary format, raw access to the contents is restricted by the storage location's constraints, whether it's a network file sharing or a SaaS-based service.

Unstructured data may have some degree of labelling and have some implicit structure put on it by convention or business practice, therefore there is a continuum between these classifications. A structured data schema can also be exploited, either accidentally or intentionally, to become functionally unstructured; for example, there is nothing more than convention that prevents the usage of a "customer account notes" column to contain Social Security numbers. Finally, the combination of a data schema and operational standards is what allows us to accomplish data security's enabling feature: categorization.

Both structured and unstructured data need to be classified in order for a data security system to know how to handle it.

The process of determining the amount of risk associated with data based on its possible impact on the organization is known as classification. The following levels are defined in the seminal publication "*Standards for Security Categorization of Federal Information and Information Systems*" (FIPS Pub 1991):

- **Low**: Limited impact on business processes due to a loss of confidentiality, integrity, and availability (for example, marketing or website content)
- **Moderate**: Confidentiality, integrity, and availability are compromised, resulting in substantial negative consequences for corporate operations (for example, customer information, price lists, business plans, or strategy documents)
- **High**: Confidentiality, integrity, and availability are compromised, resulting in severe or catastrophic consequences for company operations (for example, source code, banking credentials, or signing credentials) Although high-level, these categories can be utilized to impact initial access regulations in a Zero Trust context.

We'll go over these classifications and their implications. Zero Trust will be discussed later in this chapter, as well as in the policy model chapter.

Data lifecycle

Data, like an identity, has a defined lifespan. The lifetime of data begins with its production, continues with its use, and concludes with its annihilation. Each of these levels necessitates distinct security techniques and procedures.

Data creation

Data may be generated in a number of ways, and the method used to create it defines whether the data is structured or unstructured. Data can be stored as a file or as a record in a database, as shown in *figure 13.1*. Furthermore, data is not necessarily generated by a person or user—it may be created by an application or a process, in either a structured or unstructured manner. Business files (for example, papers or spreadsheets), machine-generated data (for example, sensor data or computed results of analysis), or valuable IP (for example, source code, equipment designs, or genetic or pharmacological data) are all examples of data:

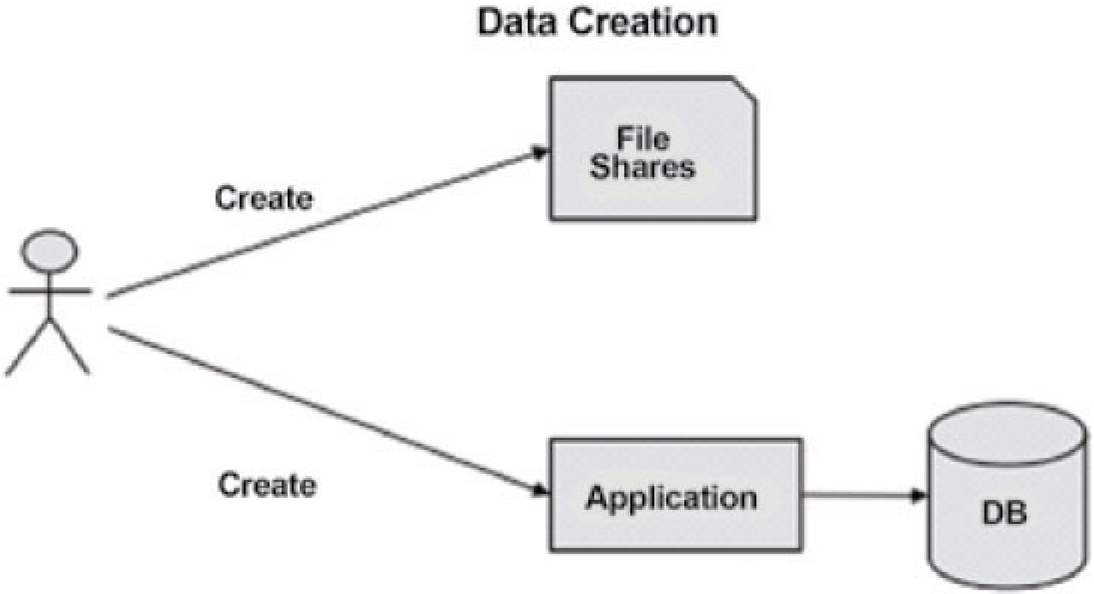


Figure 13.1: *Data Lifecycle—Creating Data*

Metadata or tagging is required to enable categorization policies, regardless of how the data is generated. These categorization tags or labels can be created in a variety of ways, including automated, user-based, and discovery solutions. Automated data categorization is when software analyses and categorizes documents using a variety of methods, such as content analysis, document location, user department, or the application or business process. This categorization is frequently done when the data is being created. Training and subject matter expertise of the data's content are required for user-based categorization.

Even with training, users can be an effective method for providing tagging and labelling, but there is a danger of inconsistency since people may apply tags and labels differently.

Finally, discovery tools categorize data, but unlike automated classification systems, they usually run after the data has been generated and saved. They apply tagging and labelling based on content, location, and search parameters, but they may not be aware of the person, application, or process that originated the data, unlike automatic categorization systems.

Data usage

While all data should be protected, categorization allows for more effective security throughout the next phase of the data's lifetime, when it is actually used. In terms of data utilization, there are three stages to consider: data at rest, data in motion, and data in use. All of these stages present both obstacles and opportunities in terms of data management and security. The *figure 13.2* shows an example of how data might travel through several phases when accessed by a user using a web browser application. The data is in the data-at-rest state before it is accessed. After the data has been produced and written to some type of persistent storage, this step begins. Full-disk or database table encryption (or another holistic solution) provides a level of security for data-at-rest, but it's vital to note that this doesn't safeguard the data as a resource. Encrypting the whole disc, also known as database table encryption, protects data from physical or disk-level access, but it is not part of an authorized paradigm:

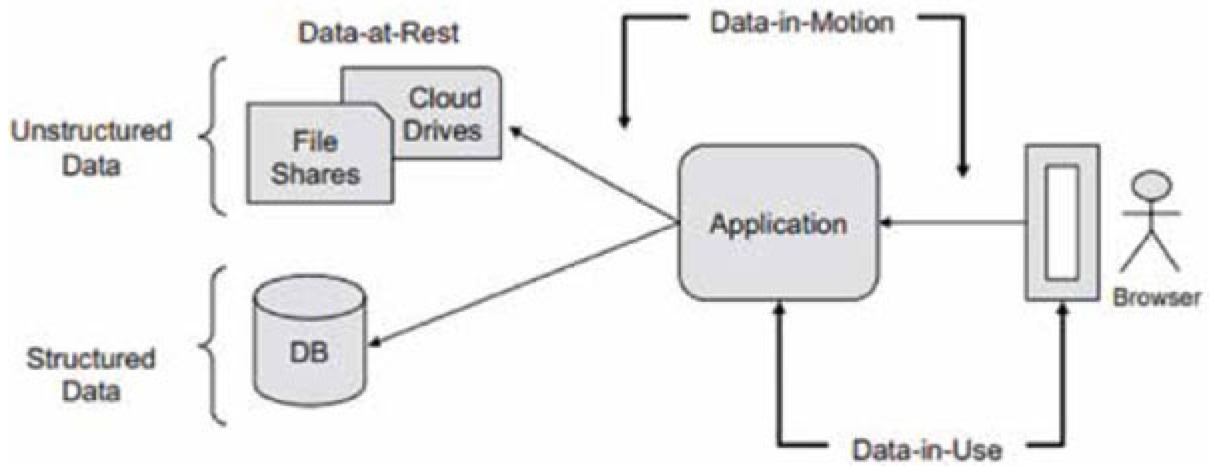


Figure 13.2: *Data Usage*

In the example shown in *figure 13.2*, there are two instances of data-in-motion when the user visits the application. To get the data, the application will call the storage location; this is the first chance to safeguard data-in-motion through an encrypted network connection between the application and storage.

A second option for protecting data-in-motion is the network between the user device and the application, which should employ HTTPS or similar secure TCP channel. In many ways, data-in-motion is the easiest phase to protect; it may be addressed simply by utilizing an encrypted network protocol, which should be used for all data-in-motion, independent of categorization.

Finally, when data is actively retained in memory within software such as application clients, browsers, or application servers, it is referred to as data-in-use. Data in this stage is frequently the most difficult to safeguard. As seen in *figure 13.2*, once the data has been loaded into memory, safeguarding it might be challenging.

There are data security techniques, such as in-memory encryption, data tokenization, or obfuscation, which can be used to protect the data in use. This is heavily reliant on the application and technology. Solutions like CASBs can help with SaaS applications, but enterprise-built apps will often rely on developers to use established design patterns and toolkits or libraries.

Data destruction

Data destruction is the last stage of the data lifecycle. Data retention rules, which specify how long data should be held and accessible before it is erased, must be defined and enforced by organizations, particularly those in regulated sectors or those that manage sensitive data. It’s important to note that different business verticals have varied needs, which can make administering “end of life” policies difficult, especially for bigger or multi-industry companies.

Data storage and retention policy enforcement are being offered as a service by a rising number of data lifecycle service providers, often via SaaS. These SaaS systems may help by enforcing categorization criteria consistently and simply, lowering the cost and labor of traditional on-premises storage and management programmes.

Various stages of the data lifecycle require various approaches to data protection. Data may be reasonably readily secured for data-at-rest (through complete disc encryption) and data-in-motion, as indicated in the preceding section (via encrypted transmissions). However, data-in-use is the more difficult and intriguing phase, which **Data Loss Prevention** (**DLP**) solutions may help with. DLP systems, which are commonly used by businesses, offer a set of technological controls that are usually centered on the following elements:

- **Device control**: The ability to determine how data may be used at the device level (for example, restricting the ability to print or copy-and-paste, or determining whether a device's USB ports can be used).
- **Content-aware control**: Enforcing and fine-tuning data security restrictions depending on the data's content.

Data obfuscation is one example:

- **Enforced encryption**: Encryption of data at rest at the disc or physical storage level. Its goal is to keep the data on the device inaccessible even if it is lost or stolen.
- **Data discovery**: One of the most critical parts of material security, discovery solutions allow businesses to not only identify but also automate the classification of unknown sensitive data.

In a Zero Trust environment, DLP solutions will continue to be useful because they offer the technological capabilities to implement access control regulations. Of course, the company must design, validate, and curate the policies that DLP systems enforce. These tasks fall under the heading of **Data Access Governance** (**DAG**), which is a subset of information security.

DAG is directly tied to IAM's identity governance capabilities, and it determines where and how data may be accessible, as well as who can access it and when. Using DAG to describe the constraints under which data may be accessed in a Zero Trust environment should ideally be related directly to Zero Trust rules. DAG offers capabilities and access rules that regulate data and, as a result, how policies are implemented across the company.

Data governance may successfully offer a method for access regulations to be implemented and further maintained through metadata tags through data classification. Zero Trust RBAC or ABAC rules can use these metadata tags as input.

Digital Rights Management (**DRM**) is another sort of data security mechanism that gives owners of proprietary data, data with a copyright attached, or any other type of company data that might be important intellectual property management (IP). DRM establishes technological limitations established by the data owner and can limit how that data is used and accessed in the short and long term. Some DRM systems can and do integrate with Zero Trust platforms, taking use of contextual information like identity and device details. While DRM focuses on restricting access to data, other measures such as classic data encryption, newer approaches such as data tokenization, and upcoming technologies such as homomorphic cryptography2 all obfuscate the data itself, allowing Zero Trust principles to be implemented. Regardless of the obfuscation approach, integrating Zero Trust into these technologies can allow identity and context-aware data access controls. In the next part, we'll go deeper into this topic.

Zero Trust and data

We discussed numerous Zero Trust deployment strategies in Chapter 3, each of which enables PEPs to secure resources. These resources were purposefully depicted generally in these scenarios—they may be data, programmes controlling/modifying data, or transactions. In every scenario, the Zero Trust PEP employs policies to safeguard these assets, and as previously stated, PDPs must make access decisions based on contextual data. When it comes to data, Zero Trust regulations can make use of categorization and metadata. So, in a Zero Trust environment, let's take a look at what a PDP and a PEP look like in terms of data.

Data categorization is accomplished in the environment by labelling and tagging data pieces, as previously indicated. If at all practicable, Zero Trust regulations should incorporate this labelling and tagging. These policies should also contain access decisions based on data properties and give access based on roles, attributes, or other identification data.

While these classifications and regulations may be based on the organization's risk model and risk tolerance, the actual controls implemented by the security system must be based on the risk model and risk tolerance.

To elaborate on this idea, an organization's audit and security team will usually create controls to satisfy regulatory and compliance standards. **Sarbanes Oxley** (**SOX**) regulations apply to publicly listed companies in the United States, for example.

Because SOX requirements focus on data, classification through tagging and labelling will enhance audit and security teams' rules as they apply to financial data. A data access governance system may be used to apply some of these regulations and give certain capabilities:

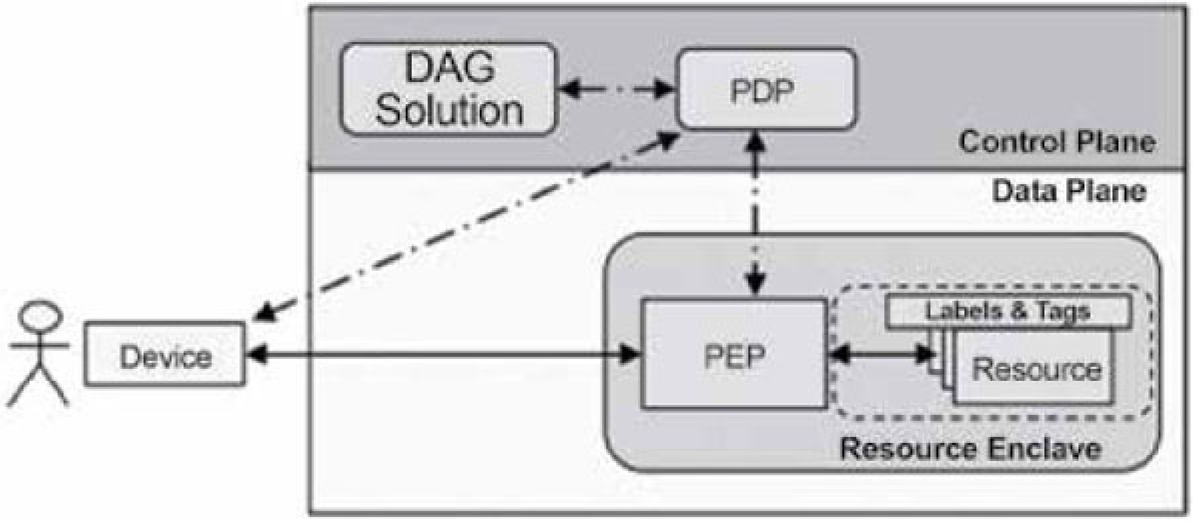


Figure 13.3: *Data Management in the Enclave-Based Model*

The *figure 13.3* shows how a data security solution might be implemented in the Zero Trust enclave architecture. The data being secured by the PEP, which employs policies set by the PDP with input from a DAG solution, is the resource in this architecture. The access being safeguarded in this picture might be direct access from the user's device or an application accessing the data on the user's behalf. If the whole data resource is categorized as *Customer Records*, for example, only identities belonging to a certain group (Customer Care Team) should be able to access it. The PEP may prevent an application attempting to access this data from outside the resource enclave as a result of this policy enforcement. In this case, the application may need to have a Zero Trust identity that can authenticate and give identity context in order to acquire access. There would be bidirectional connectivity between the PEP and the data management system or application in an efficient Zero Trust solution. The data management platform would give the PDP and PEP data elements to utilize in policy choices and enforcement. Furthermore, the data management system would be able to receive contextual data from the Zero Trust system in order to execute real-time policy enforcement measures. Even if some data is kept locally on user devices, it must still be protected. The *figures 13.4* and *figure 13.5* show how Zero Trust may be used in combination with data security technologies in two distinct ways:

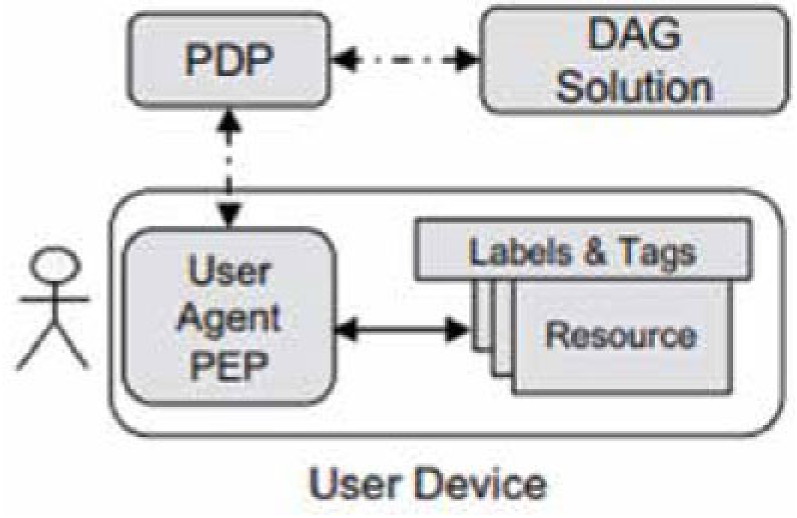


Figure 13.4: *Data Access Governance and Data Protection on a User Device*

The *figure 13.4* depicts how a Zero Trust system may be used in conjunction with a Data Access Governance solution and the user agent PEP on the user’s local device. The DAG system gives input into the PDP since DAG solutions specify policies rather than actively enforcing access constraints. This extra information should educate the PDP about data regulations and assist the PEP in enforcing access restrictions based on the data’s labels and tags on a local level:

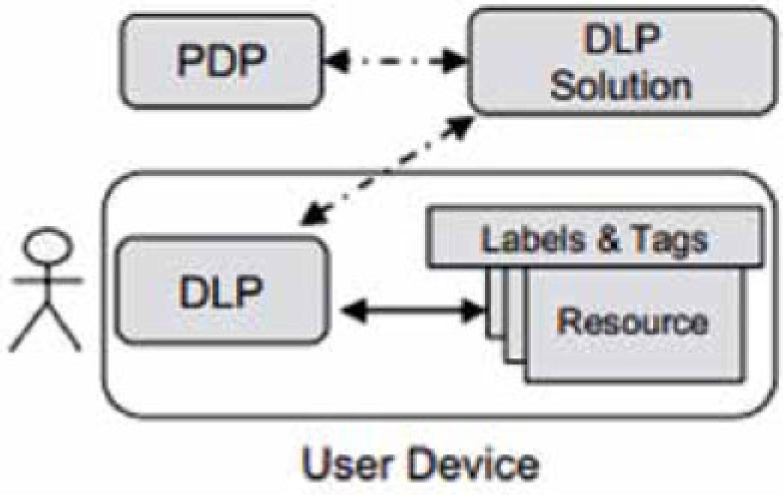


Figure 13.5: *Data Loss Prevention and Data Protection on a User Device*

In contrast, *figure 13.5* depicts how the user's device contains a DLP component that actively enforces controls. In this case, the Zero Trust system provides the DLP system with identity and session context information for use in its internal authorization (access control) model. A Zero Trust system, for example, may supply user geolocation data, allowing the DLP to enforce data residency rules. As a result, the local DLP mechanism basically becomes a (mini) Zero Trust PEP.

Conclusion

Preventing Data Loss (*figure 13.5*). We've focused on data as a resource in the Zero Trust environment throughout this chapter, which, like other resources, requires security. Data must be accessible through PEPs, which impose an identity-centric security environment, according to our Zero Trust viewpoint. Data lifecycle management, data governance, and data loss prevention are all vital aspects of data security that will continue to exist (and be successful) even if a Zero Trust solution is not implemented. A data security solution will be improved by using an identity-centric security approach. This is, however, a far more complicated case.

Although it's not the greatest candidate use case for an early project, we recommend that your Zero Trust approach incorporate context sensitive data security at some point. This will be reliant on the data security capabilities of your selected Zero Trust platform. On a User Device, and Data Protection.

CHAPTER 14

Infrastructure and Platform as a Service

Cloud computing adoption has been one of the most significant and important trends in our business in the last decade, and it shows no signs of slowing down. IaaS and PaaS services have revolutionized the way we build, deploy, and access software. We do not think, however, that these platforms have had a same broad and meaningful influence on security. While these platforms feature complex and strong access control models, they're primarily geared to safeguard services hosted in their cloud environments, rather than serving as comprehensive corporate security solutions for users across many heterogeneous settings.

This broad scope—ensuring access to all resources for all users—is, of course, a core Zero Trust premise. This isn't to say that these IaaS and PaaS cloud platforms can't be part of a Zero Trust security deployment (even if only a small component). After all, Google developed many of these concepts internally and has begun to commercialize some of them as part of its cloud platform. However, the security solutions offered by the main cloud suppliers are primarily focused on providing security within their cloud platforms, rather than offering general-purpose protection throughout the company. Microsoft is an exception, since it is exploiting its strengths in identity, desktop operating systems, and cloud computing in some novel and exciting ways. Remember that our goal is to provide you with a framework and a set of tools so that you can make thoughtful and informed decisions about how to best proceed with your Zero Trust initiative. Our goal is not to evaluate or rank vendors and their offerings—a that's highly dynamic and moving target—rather, our goal is to provide you with a framework and a set of tools so that you can make thoughtful and informed decisions about how to best proceed with your Zero Trust initiative. IaaS and PaaS are so vital in today's businesses that any Zero Trust programme would almost certainly incorporate them. Let's get started.

What is it?

Infrastructure as a Service (IaaS) is well-known and easy to define dynamic provisioning of a whole operating system in a **Cloud Service Provider** (**CSP**) environment with a "*pay as you go*" service model. Enterprise customers are in charge of setting and maintaining the whole operating system and network, as well as installing any necessary applications on the virtual server. Enterprises are effectively leveraging infrastructure as a service to deploy and configure an OS image of their choice into a virtual "*bare metal*" computer. Platform as a Service, on the other hand, comprises a wide range of functions and models among CSPs, as well as a potentially confusing set of capabilities.

When discussing PaaS, the term "serverless computing" is commonly used to refer to the ability to deploy custom code that implements functionalities that you've created without having to install a full server operating system. Serverless functions are hosted in a PaaS environment, which includes infrastructure for accessing, maintaining, and launching them.

We know that we're excluding a wide range of PaaS features offered by the main CSPs, including cloud functions, containerized workloads, service meshes, and everything in between. Later in this chapter, when we categorize and investigate the ways in which they might be incorporated into a Zero Trust environment, we'll look at some of these.

While IaaS and PaaS have significant distinctions, they do share certain similarities. They are mostly used to hold and execute bespoke resources that the organization has built and deployed. For example, these resources may be bespoke code in a function, a whole executable programme or web application, or even just an enterprise-designed database. In every case, there are resources (code or data) that the company wants to make available to certain individuals, necessitating the use of an access control architecture.

IaaS and PaaS are crucial to Zero Trust since they account for a significant amount of how applications are generated and deployed today. CSPs, on the other hand, offer sophisticated and reliable access control mechanisms that enable parts of Zero Trust. The Identity-Aware Proxy, for example, is a Google Cloud Platform feature that enforces identity-centric remote access regulations for GCP services. When these resources are accessed by external identities, however, CSPs' internal security mechanisms impose some complication. Remote access, in particular, is frequently outside the scope of the CSP, necessitating coordination or alignment with another security solution in areas where access crosses security domain borders. A Zero Trust platform can help with this by standardizing security and access restrictions across systems and silos. Integration of Zero Trust and native CSP security, such as leveraging cloud metadata tags as input into contextual rules, is legitimate and helpful. However, you must exercise caution in attempting to do too much. It may not make sense to utilize your Zero Trust system to handle access control for services that are deployed and called solely within an IaaS or PaaS platform, even if it is technically possible. The decision of where and when to limit the scope of your Zero Trust effort will be critical to its success. Let's continue our examination of IaaS and PaaS services by looking at how Zero Trust may and should be applied in these settings.

Zero Trust and Cloud Services

Your Zero Trust deployment approach, as well as the sorts of cloud platform services you've decided to utilized, will determine how a Zero Trust security platform fits into an IaaS or PaaS environment. The Zero Trust enclave-based and cloud routed deployment methods, in particular, perform effectively since the PEP is external to the resources being protected in both situations. That is, they serve as a natural architectural component at the CSP's external access barrier, allowing the Zero Trust system to apply its identity-centric regulations before allowing users to access resources within the cloud. The resource-based and microsegmentation models, on the other hand, need two things that may be difficult to do in cloud systems. First and foremost, the PEP must be operating on the resource. This isn't a problem with IaaS resources, but it isn't always the case with PaaS resources. Second, none of these models often includes a means for implementing network-wide access control. Subjects must have direct network access to the PEPs, in other words. This works for local network services and resources, but requires a distinct access method for remote topics, which is not included in these two models. These constraints, in our opinion, make it challenging to employ Zero Trust models for IaaS and PaaS resources in many instances, especially for cloud services that are likely to be accessed from a range of locations. Furthermore, inside the PaaS context, CSPs have their own internally created (and typically extremely effective) security frameworks for service-to-service access. In reality, it's probably best to use the CSP's native access control mechanism for internal PaaS services rather than imposing a possibly incompatible external approach. We'll go over this in more detail later in the chapter when we talk about service meshes. So far, we've shown that the PEP works well as an access control point across the cloud border as we've looked at how to apply the Zero Trust security model to IaaS and PaaS services (at the ingress point into the cloud environment). Let's look at how cloud services are accessible and how access may be managed to see how this works in practice. For convenience, our explanation and graphics will be centered on the enclave-based Zero Trust paradigm, albeit they will function similarly in a cloud-routed Zero Trust system. Unlike SaaS services, which we examine in the next chapter, IaaS and PaaS platforms all come with built-in access control mechanisms, making them universally compatible with a Zero Trust PEP. There are several technological techniques by which cloud platforms impose this access control; for the sake of simplicity, we'll refer to them as an access gateway, which acts as a logical ingress firewall into an IaaS or PaaS environment and performs source IP address filtering. Although rudimentary, this capacity is all we need to achieve our goal: our Zero Trust system (enforced by the PEP) is how we implement dynamic and identity-centric rules:

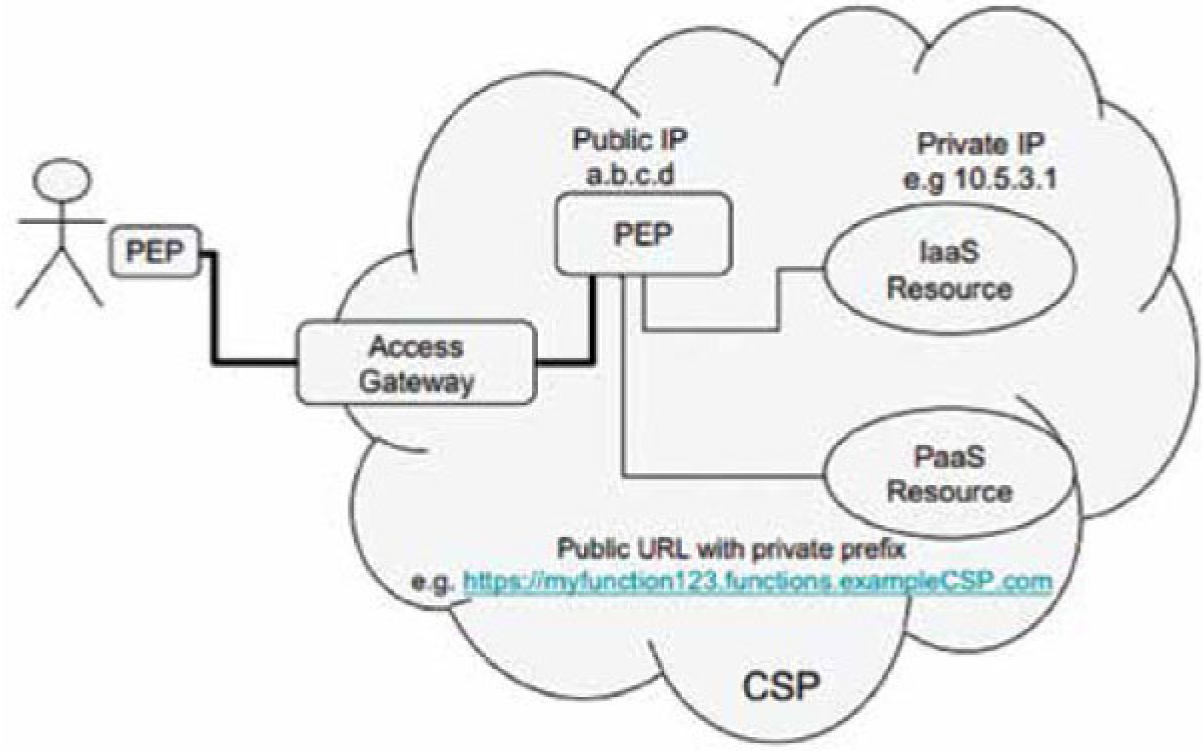


Figure 14.1: *Cloud Access Control via Co-located PEP*

The *figure 14.1* shows how a PEP operating on a CSP platform may be used to regulate access to IaaS and PaaS services within the same cloud environment. This model's access restrictions fall into one of two types. IaaS resources are given an IP address, and access to that IP address is controlled by the CSP's access gateway, allowing only traffic from the PEP to reach the resource. The *figure 14.1* depicts this in the case of an IaaS resource with a private IP address of **10.5.3.1** (to which distant users would not be able to route traffic anyhow). The access gateway is set up to allow remote access to the PEP from any external IP address, such as a device belonging to a distant user. Of course, the PEP (and the PDP, not depicted) are responsible for enforcing the Zero Trust regulations; the access gateway's primary purpose is to guarantee that all resource-bound traffic is routed via the PEP and therefore subject to its restrictions. It's worth noting that even if this IaaS resource had a public IP address, the diagram and the ultimate result may be same. The system can ensure that its Zero Trust principles are implemented as long as the CSP network is set up so that all resource-bound traffic can only originate from the PEP. Also, while displayed as a single object, this resource might really equate to a single service (TCP port) operating on an IaaS instance. This strategy may be beneficial for an IaaS instance that runs a public HTTPS web server on port 443, but also has a PEP-protected administrative SSH interface on TCP port 22.

A PaaS resource is also shown in *figure 14.1*, which is accessible using a standard cloud platform pattern: a public URL with a private identification as a prefix to the FQDN. The graphic depicts a notional example, https://myfunction123.functions.exampleCSP.com, but real-world examples include https://abc123def.execute-api.us-east-1.amazonaws.com for an AWS lambda function and https://myapp1.azurewebsites.net/api/myfunction123 for an Azure function.

There will be a set of public IP addresses shared by many, many functions in these cases, with the CSP infrastructure providing load balancing and mapping to a specific customer's account. These IP addresses, as well as the computation and network infrastructure that serves them, are under the CSP's control and cannot be controlled or tampered with by a single client. But that's fine; it doesn't go against our Zero Trust security paradigm. This is because, despite the fact that the IP address and the actual network entry point are both public, CSPs allow you to limit the source IP addresses that can call a certain function. Of course, in this case, we'd just set it to allow access only from the PEP. This is an example of how to use some fundamental cloud features, such as source IP address constraints, to enable the Zero Trust security paradigm. Finally, because the PEP is local to the cloud platform, it may perform API calls to collect metadata about the resources in the local cloud environment, which it can use to choose targets (resources) for rules given to it. Similarly, the local PEP may detect new service instances across business cloud accounts and dynamically (and automatically) offer the appropriate degree of access to the appropriate distant users. Of course, in this case, we'd just set it to allow access only from the PEP. This is an example of how to use some fundamental cloud features, such as source IP address constraints, to enable the Zero Trust security paradigm. Finally, because the PEP is local to the cloud platform, it may perform API calls to collect metadata about the resources in the local cloud environment, which it can use to choose targets (resources) for rules given to it. Similarly, the local PEP may detect new service instances across business cloud accounts and dynamically (and automatically) offer the appropriate degree of access to the appropriate distant users. Detecting new resources and analyzing resource properties in this manner is a key function that PEPs may provide for cloud settings, as we'll see in our forthcoming chapter on policies:

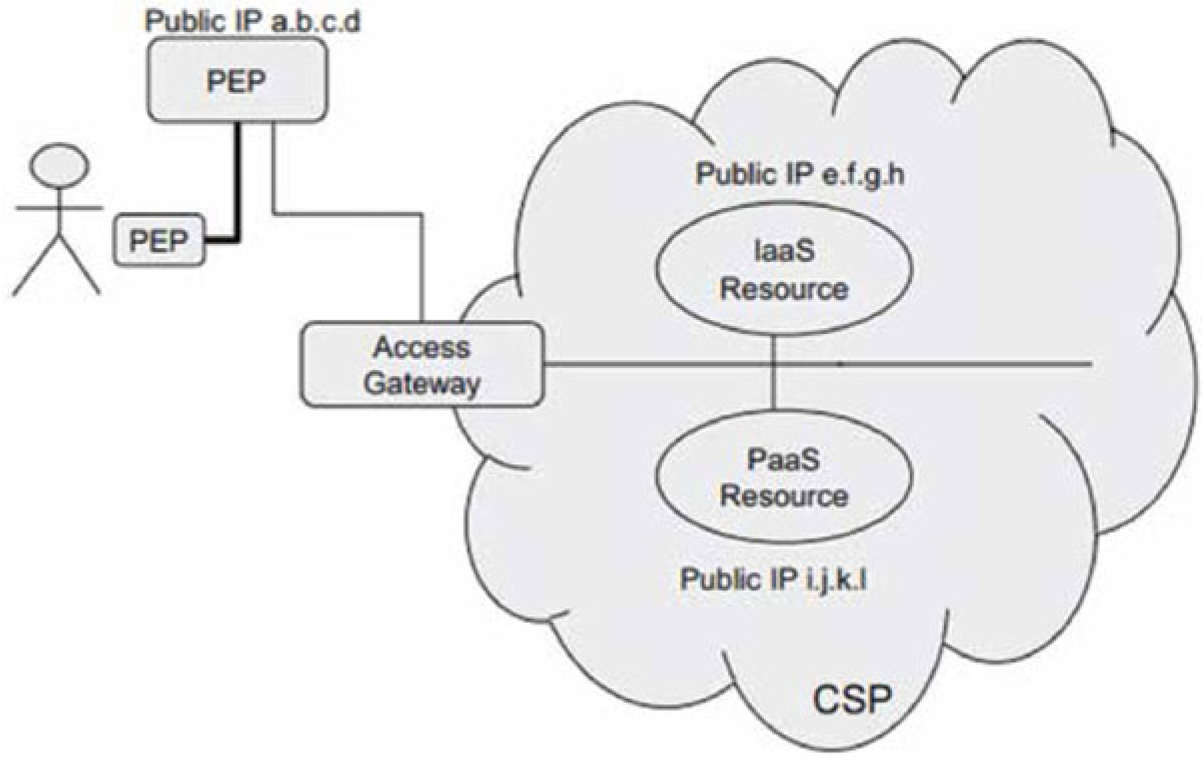


Figure 14.2: *Cloud Access Control via Remote PEP*

The figure 14.2 shows how to impose access control to CSP-based resources using a remote PEP operating in any environment (on-premises or in another cloud environment, it doesn't matter). These resources might be IaaS or PaaS, as in our prior instances, but they must have a public IP address because they are being accessed remotely in both cases. We're leveraging the CSP's basic capabilities to impose a source IP address limitation for these resources, mandating that all traffic for these resources come from the PEP's public IP address. This simple solution, like the last example, allows us to apply our Zero Trust model's identity-centric and dynamic access restrictions to our CSP-based resources. It's worth noting that because this architecture uses the native application protocol from the PEP to the access gateway and then to the Resource, it's only suited for encrypted communications. Of course, CSPs offer a wide range of network and security features, including network security groups and IAM rules, that go beyond the general access gateway we've explored here. At a minimum, these may be used together to restrict access to resources (services) according on their originating IP address, ensuring that they can only be accessed from a Zero Trust PEP. This is the essential skill for incorporating cloud resources into a Zero Trust environment. We have (intentionally) depicted the network topology in a simplified manner in order to demonstrate the principles; real-world cloud platforms give numerous methods to connect cloud services into your company networks. CSPs, for example, often provide a “direct connect” site-to-site VPN architecture, which logically extends an on-premises network to a private cloud network via a local telecoms provider. CSPs also provide more advanced network connectivity and configuration capabilities, allowing you to create complicated network topologies and access control systems. However, we propose that you keep things simple and externalize your Zero Trust platform’s dynamic and identity-centric access restrictions. This is what Zero Trust excels at, and it lets you avoid creating a new, complicated, and CSP-specific security architecture. While strong, CSP approaches are more network and IP address oriented than identity centric. They also lack the capacity to establish and enforce the sorts of Zero Trust policies that we require across our heterogeneous and varied corporate settings. Of course, there are exceptions to any rule, and we recognize that forcing your Zero Trust system into every aspect of your environment is neither practicable nor acceptable. In fact, knowing where to create boundaries is an important element of a successful Zero Trust journey. Finally, you must guarantee that each aspect of your environment has the most suitable and effective security platform, tools, and processes. The service mesh, which is a framework for delivering and managing containerized workloads in a reliable and scalable manner, is an excellent example of this. In certain sense, service meshes constitute a self-contained Zero Trust microsegmentation concept and system. Let’s look at how they function and how you might integrate them into your larger business System of Zero Trust.

Service meshes

Service meshes are a relatively new and fast gaining popularity method of delivering containerized applications at scale. While the open source meshes may undoubtedly be installed on premises, we’ve seen them utilized most in cloud environments. Service meshes, like as Istio and Linkerd, are ideal for developing current DevOps-style microservices-based applications. Service meshes are systems for deploying, controlling, and managing large-scale containerized (microservices) workloads, with a focus on microservice communication. For example, according to the Istio documentation, “Istio provides automated baseline traffic resilience, service metrics collection, distributed tracing, traffic encryption, protocol upgrades, and advanced routing functionality for all service-to-service communication without requiring changes to the underlying services.” What’s remarkable about these techniques is how they use a configuration-based platform to deliver a comprehensive set of deployment, communications, and runtime capabilities to microservices. This frees developers to focus on business logic rather than infrastructure, similar to the promise of application servers (App Servers) from the late 1990s. Of course, technology has advanced since the 1990s, and the service mesh approach to security has changed as well. Let’s examine the underlying structure of a service mesh (we’ll use Istio as an example) to understand how it relates to the Zero Trust microsegmentation concept. The *figure 14.3* depicts the high-level Istio mesh architecture, which we’ll examine from a Zero Trust security standpoint:

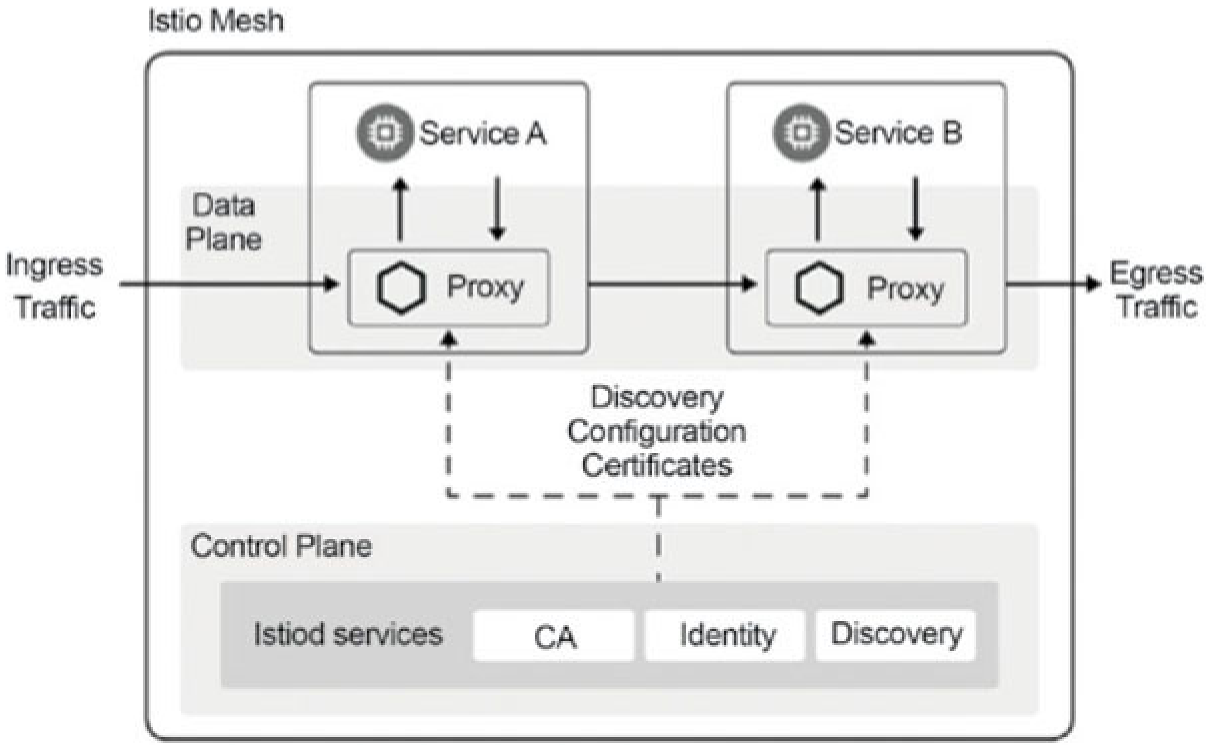


Figure 14.3: *Istio Architecture*

The separation between the control plane and the data plane, as well as a series of distributed proxies, one in front of each service, are the first things to notice. Unsurprisingly, these proxies serve as **policy enforcement points** (**PEPs**). The istiod services are the control plane's **policy decision point** (**PDP**), operating as the system Certificate Authority, managing service identification, and storing and assessing authentication and authorization rules, among other things. The proxies guarantee that service-to-service communication takes place via a mTLS channel, which ensures secrecy and authentication for both service consumers and providers.

The Istio security architecture is built on a declarative policy mode that determines which policies apply to which services based on service properties (such as namespaces and labels) as subject criteria. The proxy (PEP) in the authorization model evaluates requests based on the requestor's characteristics, the destination service's attributes, and the request metadata and header information. It's worth noting that within the mesh, requestors and services are addressed by service IDs rather than IP addresses—in fact, in many circumstances, these services all have the same IP addresses, thus IP addresses are no longer a useful way to distinguish them. We've only given a quick overview of service meshes and their security models here, but it should be enough to persuade you that they are well-designed platforms with internal security policy and enforcement processes (granted, with varying degrees of support for identity-centric and context-based policies). Service meshes are an excellent example of security solutions with enough of their own "center of gravity" to justify their usage in the company, even as part of a larger Zero Trust programme. In contrast, services like IaaS often have rudimentary network-centric security measures and must be safeguarded through the enterprise's Zero Trust platform rather than the cloud-native paradigm. Fortunately, service meshes set a clear boundary for their scope—the mesh's edge—and can quickly and efficiently enforce ingress and egress regulations using a Zero Trust platform. Service meshes are especially well-suited for integration with Zero Trust systems that rely on external PEPs, such as the enclave-based and cloud-routed models. From the standpoint of the Zero Trust system, the mesh becomes the implicit trust zone in these instances. What we outlined (and what is entirely doable today) is essentially a deployment of a wide enterprise Zero Trust system with a service mesh from a distance. It'll be fascinating to see, perhaps in the not-too-distant future, a Zero Trust solution in which the PEP may render policies based on workload parameters within the container context, limiting external access to containerized workloads. This would simply necessitate a simple mechanism to describe container workload attributes in the Zero Trust policy model, as well as a way to broadcast access control decisions to the mesh for enforcement. Some of this infrastructure already exists; for example, transmitting Zero Trust context into Istio through HTTP request headers is now doable. As firms' Zero Trust initiatives progress to greater degrees of maturity, this form of integration will become more attractive and beneficial.

Conclusion

It's apparent that IaaS and PaaS will only expand in prominence and influence on corporate application development and deployment in the future. These platforms have significantly increased the breadth and depth of their capabilities, including the ability to execute certain cloud-managed services on-premises. This is owing to pervasive network connectivity as well as incredibly low-cost computing and memory. Even in the last several years, this, paired with advanced control software, has resulted in a growth of what we term "as a Service" offers. With major CSPs developing in this area, the idea of putting service-based (and cloud-managed) computing or sensor nodes directly into a corporate network is becoming more prevalent. The term "*fog computing*" has been used to describe this tendency.

It will be fascinating to observe how these solutions and architectures evolve in terms of security—clearly, distributed computing elements will require distributed security, and there will be opportunities to combine them with corporate and CSP-based Zero Trust systems. Enterprise application architectures are also changing swiftly to take advantage of new IaaS and PaaS capabilities, and security teams must not only keep up, but also lead and support this. We think that the best approach to do this is through a Zero Trust architecture and platform.

The principles and concepts covered in this chapter should help you understand how to secure IaaS and PaaS installations as part of your Zero Trust project, as well as how to enable service mesh-based applications.

The following chapter looks at SaaS apps to round up our look at Zero Trust and cloud-based systems.

CHAPTER 15

Software as a Service (SaaS)

Of course, cloud-based software as a service (SaaS) is an important part of today’s IT and business landscape, and it has had a significant influence on how commercial software is developed and consumed. This move has made sophisticated business software a lot easier to use, with companies being able to join up, create an account, and start getting value in minutes. SaaS is defined as a publicly available web application in which the service provider (vendor) hosts, controls, and maintains the infrastructure while the subscriber performs (and manages) the prescribed application function through the Internet. For efficiency, SaaS systems are usually multi-tenant, with each subscriber having access to only their personal data. We can instantly identify some significant distinctions between SaaS and the IaaS/PaaS resources we mentioned in the previous chapter from the standpoint of Zero Trust security. To begin with, SaaS apps are designed to be publicly available, with any user on the Internet being able to access them via an HTTPS connection. That is, by definition, a SaaS system’s entry points are public rather than private. They can also only be accessed over an encrypted connection. This implies that a PEP isn’t required for resource concealment in SaaS apps (since it’s a non-goal for SaaS), and network traffic isn’t encrypted (because it’s already utilizing HTTPS). This naturally begs the question of whether or not Zero Trust is still applicable to SaaS resources, and if so, how. Even while we admit that Zero Trust accomplishes fewer things for SaaS resources than it does for private resources, we feel that utilizing Zero Trust to monitor and regulate access to SaaS apps provides benefit. Zero Trust may implement identity-centric and context-sensitive access controls even for publicly available SaaS apps. Because the PDP is connected with identity providers and other corporate systems, it may regulate access to public resources by using group membership, as well as identity, device, and overall enterprise system attributes.

While many (but not all) SaaS apps can and do interact with identity providers for authentication, they generally do not manage access using device, identity, or system attributes.

Of course, SaaS application security encompasses more than simply access control, and the security industry has developed an ecosystem of SaaS-specific security products, such as **Secure Web Gateways** (**SWG**) and **Cloud Access Security Brokers** (**CASB**). Let's look at each of them and see how they connect to Zero Trust.

SaaS and Cloud Security

To discuss Zero Trust with SaaS, we must first look at the fundamental components of cloud security. We'll start with native SaaS security controls before moving on to the Secure Web Gateway and Cloud Access Security Broker sections.

Controls for SaaS that are native to the platform

Despite the fact that their solutions are open to the public, SaaS providers realize and accept the necessity for some level of access and network security. Of course, they've put in place procedures to safeguard their services from Internet-based attacks like DDoS, as well as internal systems to ensure the platform's integrity and availability. Furthermore, many SaaS platforms have two built-in access control techniques for businesses. The first is the ability to implement source IP address limitations, which is the same core network access control capability as IaaS and PaaS systems. The second is federated identity management, which involves the SaaS system delegating user authentication to a third-party identity provider. Let's take a look at each one separately.

SaaS systems must implement source IP address limitations differently from IaaS/PaaS platforms, in that they only enforce the source IP address constraint for users linked with a certain account. The https://MySaaSApp.com/login website, for example, is accessible to anybody on the earth, but the SaaS platform will only allow users from the mycompany.com domain to join in if their traffic originates from the defined IP address. This is an authentication access control policy that is applied to a customer’s tenancy on the SaaS platform. This feature may be used to demand access via a regular VPN or a Zero Trust solution, which both route user traffic through an enterprise-controlled network with a known IP address egress point. Federated identity management is when an application uses standardized technologies like SAML and OpenID Connect to connect to an external identity provider for user authentication. This is an effective technique to implement identity-centric features of Zero Trust access restrictions that is (interestingly) independent of network-level security. Users cannot authenticate directly into the SaaS application from a technological standpoint; instead, the SaaS app either receives a current authentication token from the user’s browser or redirects the browser to the identity provider for authentication. This, of course, makes use of any authentication factors and contextual controls the identity provider has set up. Keep in mind that this is usually only used for authentication purposes. Internal authorization models, in which users are allocated to various roles that regulate their access within the programme, are still widely used in SaaS systems. Most SaaS apps do not yet have ways to ingest external contextual information and make authorization choices based on it—this is a more sophisticated and forward-thinking use case that we’ll discuss again in the summary of this chapter. Finally, there’s no reason why these two techniques can’t be used together—for example, employing a federated identity system for authentication and a Zero Trust network solution for deep device posture checks. Following that, we’ll look at CASBs and SWGs, two key aspects of cloud security that businesses employ to get insight and control over user access to SaaS apps. Surprisingly, there’s more overlap and convergence between these formerly discrete market categories, which is part of a wider trend toward the consolidation of a broad set of network and security tasks into a single service offering (the Secure Access Service Edge).

Secure Web Gateways

Secure Web Gateways, which may be installed on-premises or as a cloud-based service, allow businesses to regulate which websites their employees can visit while also providing antimalware and threat protection. TLS termination is often performed by SWGs, which function as a man-in-the-middle web proxy to check communication contents. Endpoint agents are used by some SWGs to assist in the acquisition of Internet-bound traffic (as well as to provide extra services). On-premises business SWGs are becoming less common, with cloud-based SWG services taking their place. In some aspects, the SWG policy model is the polar opposite of the Zero Trust approach, since it is designed to deny access to banned Internet sites rather than simply allowing access to clearly approved locations like the Zero Trust model does. That is, most SWGs function in a "*default allow*" mode, which makes sense in most circumstances given the vast number and variety of websites available on the Internet. SWGs are generally connected with business identity providers for user authentication, and they can impose multiple access control policies based on factors like group memberships. SWGs, on the other hand, do not provide network security or remote access to private resources on their own—this is just not in their scope of business. Some cloud-based SWG providers have expanded their service offerings to incorporate Zero Trust-style access management to private resources, as we'll explore in the following paragraph.

Cloud Access Security Brokers

CASBs are commonly utilized by businesses to address the "*shadow IT*" problem, which occurs when business teams begin to use SaaS-based apps outside of IT's visibility and control. CASBs address this issue by detecting and reporting on SaaS application usage, as well as offering a set of risk and compliance assessment capabilities. They also provide value by imposing DLP restrictions on SaaS-based data, and they may frequently combine user identification and device-based access policies, usually in combination with SAML or OpenID Connect-based identity providers. It's intriguing to consider the role of CASBs in enforcing adaptive and risk-based authentication and authorization based on identity and device attributes. They look to be operating as Zero from this vantage point. They do provide network security services, but their enforcement approach is centered on SaaS apps, thus they don't provide network security functions. CASBs are also not designed or developed to offer access controls for private or on-premises apps, therefore these functionalities are not included in their policy and enforcement frameworks. Companies who started in the CASB field have expanded their platform capabilities into other functional areas, similar to the SWG vendors we described before. After we talk about Zero Trust and SaaS, we'll talk about industry convergence.

Zero Trust and SaaS

Whether your security architecture contains a SWG, a CASB, or none, it should be obvious that Zero Trust security can be applied to and operate effectively with SaaS services. As long as the SaaS platform allows for source IP address restrictions and the Zero Trust system is capable of defining SaaS applications as targets within the policy model and capturing traffic bound for them, Zero Trust security systems can provide identity and context-sensitive access control to SaaS applications. Even when used in combination with a Zero Trust system, SWGs and CASBs will be valuable, but businesses must be aware of how these distinct systems operate with and influence traffic and network routing. Enterprises might, for example, use their Zero Trust system to manage access to private resources alone, while implementing a SWG and/or CASB for their SaaS apps. This is a sensible strategy.

Zero Trust and edge services

A converged and cloud-based network and security solution that integrates several of these services is now trending in the market. This is known as the Secure Access Service Edge (SASE) by Gartner, and the Zero Trust Edge by Forrester. These terms effectively explain how cloud-based security and network providers have merged several functional offerings into a single platform-as-a-service. Networking (SD-WAN, WAN optimization, and QoS, for example) and security (firewall, IDS/IPS, SWG, CASB, DNS filtering, and Zero Trust Network Access) are common capabilities within this platform. SASE and ZTE have undoubtedly experienced significant recent expansion in terms of enterprise awareness and interest, and there has been equivalent action in terms of greater vendor marketing support, innovation, and industry consolidation (acquisition). When you take a step back, you'll notice that these convergent systems have three primary sets of functions:

- Access to private resources (Zero Trust Network Access, or ingress access)
- Network connectivity
- Security for Internet access (egress access)

The **Zero Trust Network Access (ZTNA)** element of this is very fascinating and unique in our opinion. This is because ZTNA will continue to require that elements (PEPs) be deployed into enterprise-controlled environments, such as on-premises enterprise networks, data centers, and public cloud-based IaaS and PaaS environments, even as network management and Internet traffic analysis and security are moved to the cloud. This is because of two factors.

TCP/IP networks, for example, require a local node to operate as one end of the encrypted network tunnel and to broker or proxy remote connections to private network resources. Second, the local node is necessary in order to receive and use context and characteristics from local resources as part of access policy decision criteria (we'll go over this in further detail in Chapter 17, which is dedicated to policy models). One of the reasons we feel ZTNA shouldn't be treated in the same manner as the other SASE components is the necessity for a set of nodes on local private networks—our Zero Trust PEPs. Another reason is because one of our major concepts is that the Zero Trust identity and context-sensitive security model must be implemented for all identities' access to all resources, independent of the identity's or resource's location. Despite the widespread usage of SaaS apps and the migration of many users to work from home, enterprises still have on-premises users and resources. They must also maintain control over on-premises server-to-server access, something cloud-based services frequently fail to do. All of these factors combine to explain why Gartner, for example, distinguishes between "ingress SASE" and "egress SASE," each with its own set of criteria. In any case, this is a quickly expanding field, and we think that these emerging cloud-based security systems have an opportunity to interact with and exploit Zero Trust context, whether offered by their own platform or by integrating with another vendor's offering.

Conclusion

Consider how SaaS and Zero Trust security could be implemented in the not-too-distant future. First, we expect that identification, and hence identity providers, will stay at the core of Zero Trust (though we recognize that this isn't exactly a daring forecast). However, we believe that these providers will act as "*centers of gravity*" for user access to web apps and access control models, rather than merely authoritative directories and authentication points. Many IdPs now provide access portals with launchpad icons for accessing SaaS services, which are a good illustration of this. Currently, these portals primarily provide authentication and access, but we think that identity providers have a chance to expand the scope and power of their policy models beyond authentication to include authorization. As part of a **Just-in-Time** (**JIT**) access strategy, SaaS apps might start to enable account or role provisioning across the board, utilizing a standard like SCIM.

SCIM is simply the beginning; it will be fascinating to observe if and how a standard (formal or informal) emerges around how to convey permission. We don't think apps will ever fully externalize their authorization, which is one of the reasons why XACML hasn't seen significant acceptance in the industry.

We do think, however, that a widely acknowledged method of establishing and transmitting authenticated (and hence trustworthy) identity context to SaaS services will emerge, which they will be able to consume and utilize in ways that are acceptable for their environment. One specific example of this is JIT access provisioning. Without a sure, this will be an exciting and active area to follow as we learn more about Zero Trust-aware SaaS apps and the security, operational, and financial benefits they may provide. We believe that these features will eventually "trickle down" to non-SaaS apps, but we expect SaaS firms to lead the way due to their advanced federation capabilities and platform investments.

Chapter 16

IoT Devices

We've spent a lot of time in this book talking about how to manage access by authorized entities, such as users and servers. They both use modern devices with full-featured operating systems that support the installation of third-party software, are authenticated against an identity system, have attributes or roles for context, and are using full-featured operating systems that support the installation of third-party software. As a result, these systems are highly suited for use in the Zero Trust architectures we've been describing. Of course, these aren't the only kinds of linked devices—there are billions of them, all running on lower-capability and less expandable hardware and software platforms and referred to as **Internet of Things** (**IoT**) gadgets. These gadgets are frequently found on the same business networks as the company's most valuable assets. They're also well-known for exposing security flaws and providing an enticing attack surface, thus they should be included of any Zero Trust security architecture.

These IoT devices have a wide range of roles, footprints, and capabilities, and we've purposefully included a wide range in our discussion. We include newer connected devices as well as more traditional devices on business networks in this category of IoT Devices and "things." As an example,

- Printers
- VOIP phones
- IP cameras
- Badge readers
- Smart things, such as blackboards, light bulbs, etc.
- Medical or diagnostic devices on healthcare networks

HVAC systems 194 We also want to consider other types of devices that may be running in widely distributed locations, and located on public or cellular networks, such as

- Environmental sensors
- Remote security cameras
- Machinery or vehicle sensors or actuators

Finally, there are **operational technology** (**OT**) systems for industrial automation and management, which have transitioned to standardized and interoperable TCP/IP networking in the last 10–15 years and are increasingly being connected to enterprise IT networks. Although there are various distinctions and obstacles in OT contexts compared to IT environments, zero trust designs may be applied to them. However, in this chapter, we'll concentrate on IT and corporate networks. All of these "things" have an IP address and must communicate through the network to initiate or receive conversations. These devices are also considered to be relatively closed systems, which means that businesses cannot install arbitrary third-party software on them. Of course, this isn't true for all IoT devices; there's an increasing number of them running full-featured operating systems, usually a Linux variety, on which you can install third-party applications. You might regard these as Zero Trust subjects (That is, computing devices with identities) depending on your environment and architecture, in which case our regular approach to access control and policy enforcement applies. Alternatively, you can regard them as Internet of Things (IoT) devices, in which case the ideas and methodologies discussed in this chapter will apply. In any event, these devices are typically developed, produced, and deployed without the same level of security considerations as business IT solutions. There are hundreds of problems in consumer-grade linked gadgets, and similar defects also exist in enterprise-targeted products, particularly for specialized vertical items like medical equipment. Unencrypted network protocols, hardcoded default passwords, unremovable backdoors, network and OS vulnerabilities, difficulty or impossibility of upgrading firmware, and, for physically accessible devices, the ability for an attacker to use that proximity to gain shell access on the device are all common security vulnerabilities in these devices. These devices are vulnerable for attack and data exfiltration, and malware has used them as a base to conduct network research and migrate laterally (not to mention being a favorite weak point for red teams to exploit during pen testing exercises).

It’s worth noting that some IoT devices are part of a broader, often cloud-based contemporary system—the main cloud service providers each have their own platforms that combine device-installed and cloud-based software to provide messaging, security, and data management, among other things. These systems, such as Azure IoT, Google Cloud IoT Core, and AWS Greengrass, all include a well-designed security and communications model that is self-contained in certain aspects and includes built-in support for secure bidirectional communications (often both synchronous and asynchronous). As a result, deploying and operating them independently from your broader organization Zero Trust architecture may be totally fine. This is OK; as we’ve discussed throughout the book, your Zero Trust project does not have to cover everything. In reality, intentionally eliminating key components of your IT architecture can help you focus, move faster, and achieve success. Even if you’re utilizing a contemporary IoT platform, you need understand its networking and communications architecture so that it can coexist with the rest of your network security model. Most IoT devices, of course, operate outside of these cloud-based frameworks and should be included in your Zero Trust security architecture. We’ll look at some of the security and networking concerns connected with IoT devices in the rest of this chapter, and then see how Zero Trust systems may be used to solve these issues.

Networking and security issues with IoT devices

Due to their closed nature and sometimes-constrained communications designs, IoT devices, unlike user devices or servers, typically face some complicated management, security, and access concerns when placed onto organizational networks. To demonstrate these principles, *figure 16.1* displays a simplified depiction of a corporate network. The network is made up of wired and wireless parts, and it connects a wide range of device types. User devices and Ethernet-connected equipment, such as VOIP phones, IP cameras, printers, and access door badge readers, are all part of the enterprise’s wired network. Some user devices, printers, and other linked office equipment, such as wireless conference room displays and digital whiteboards, connect to the wireless network:

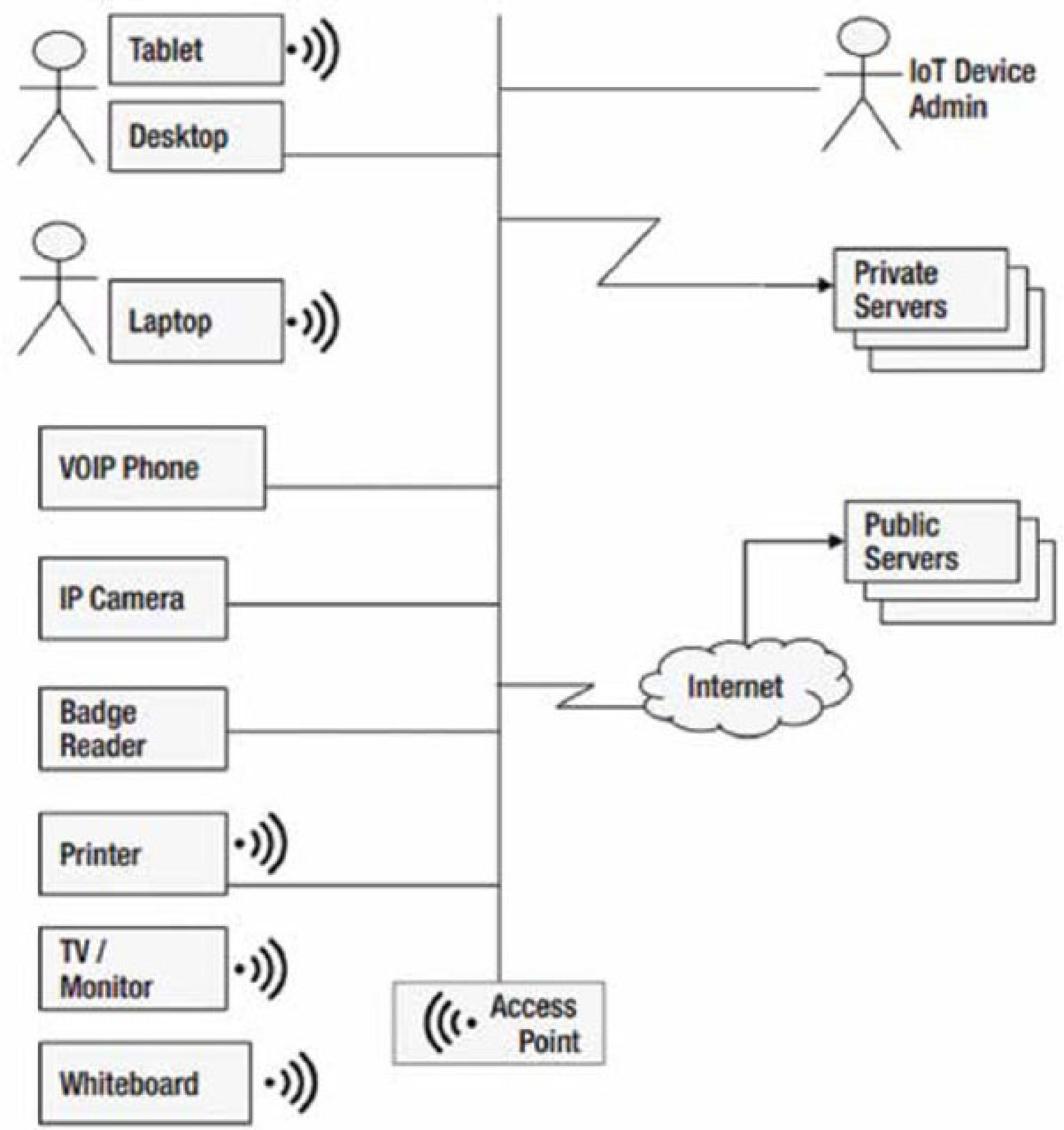


Figure 16.1: *Enterprise IoT Network*

Some devices on this network link to private servers on another portion of the company network, while others connect to servers on the Internet. There are also a few system administrators that need to link down to these devices on a regular basis to do firmware upgrades or make configuration adjustments. These administrators might work for the company or for a device manufacturer. The systems depicted in this picture often have a variety of security flaws, which we discussed briefly in the introduction. To begin with, many of these devices employ unencrypted network protocols, making them vulnerable to traffic inspection and **Man-In-The-Middle** (**MITM**) attacks, which can jeopardize confidentiality, integrity, and availability. Second, many of these gadgets contain listening ports that are exposed to the public. While this is required for our remote administrator to connect to them, it also allows any other network-connected device to connect to them over TCP. Third, these devices frequently feature insecure (or hard-coded) authentication procedures, as well as network stacks that are vulnerable to a wide range of attacks. Finally, some of these devices may be physically exposed to attackers for lengthy periods of time, such as outside environmental sensors, remote cameras, or control equipment. As a result, attackers may be able to hijack a wired network connection or physically gain access to the device (for example, by power-cycling it with a malicious USB stick inserted). When viewed through the perspective of our Zero Trust security principles, it should be evident that these solutions fall short of our key values in several ways. In an ideal world, a Zero Trust system would protect these devices by enforcing strict security policies.

- **Principle of least privilege**: Limit these devices' upstream access in the event of a device or network breach.
- **Device isolation**: Prevent unauthorized users from connecting to the devices on the network.
- **Encrypt traffic**: All native device traffic is routed over a safe and encrypted tunnel.

Of course, some of these devices (like as wall-mounted badge readers) may be connected to a separate hard-wired network, while others may be allocated to separate private VLANs to keep their traffic separate.

Those are excellent practices, but they don't (and can't) apply to all devices, and even if they did, they wouldn't render them immune to assaults. Real-world networks are sometimes clumsy, opaque, and varied, and they have frequently evolved spontaneously without a clear strategy. This is usually due to technical employees being under pressure to get things up and running as soon as possible, and not devoting enough time or funding to revise or enhance things later. As a result, mixed enterprise networks containing these devices might face a variety of issues, making them challenging to protect. For starters, these networks are often flat and open, with hundreds (or thousands) of different types of devices. This is frequently owing to the difficulties of maintaining traditional (non-Zero Trust) ACLs throughout a dispersed company, as well as keeping them up to current and in sync with daily changes. Second, unlike user devices, which are often controlled centrally, IoT devices are typically maintained either as independent devices or via a management software system that exclusively applies to devices of a specific type. As a result, configuring or managing these devices at scale can be challenging and time-consuming. Given their inability to install software, the major problem with these devices is managing their network traffic. What upstream resources are they permitted to link to, and what other network systems are permitted to connect to them? Of course, the PEP—either a network PEP, a user agent PEP, or both—plays this function in a Zero Trust system. We'll next look at how we may bring these worlds together, as well as some of the technological issues that frequently occur.

IoT devices and Zero Trust

IoT devices would be put on an isolated and homogeneous network, with a Zero Trust PEP controlling all north/south network access into the isolated network. The *figure 16.2* depicts this optimal logical situation:

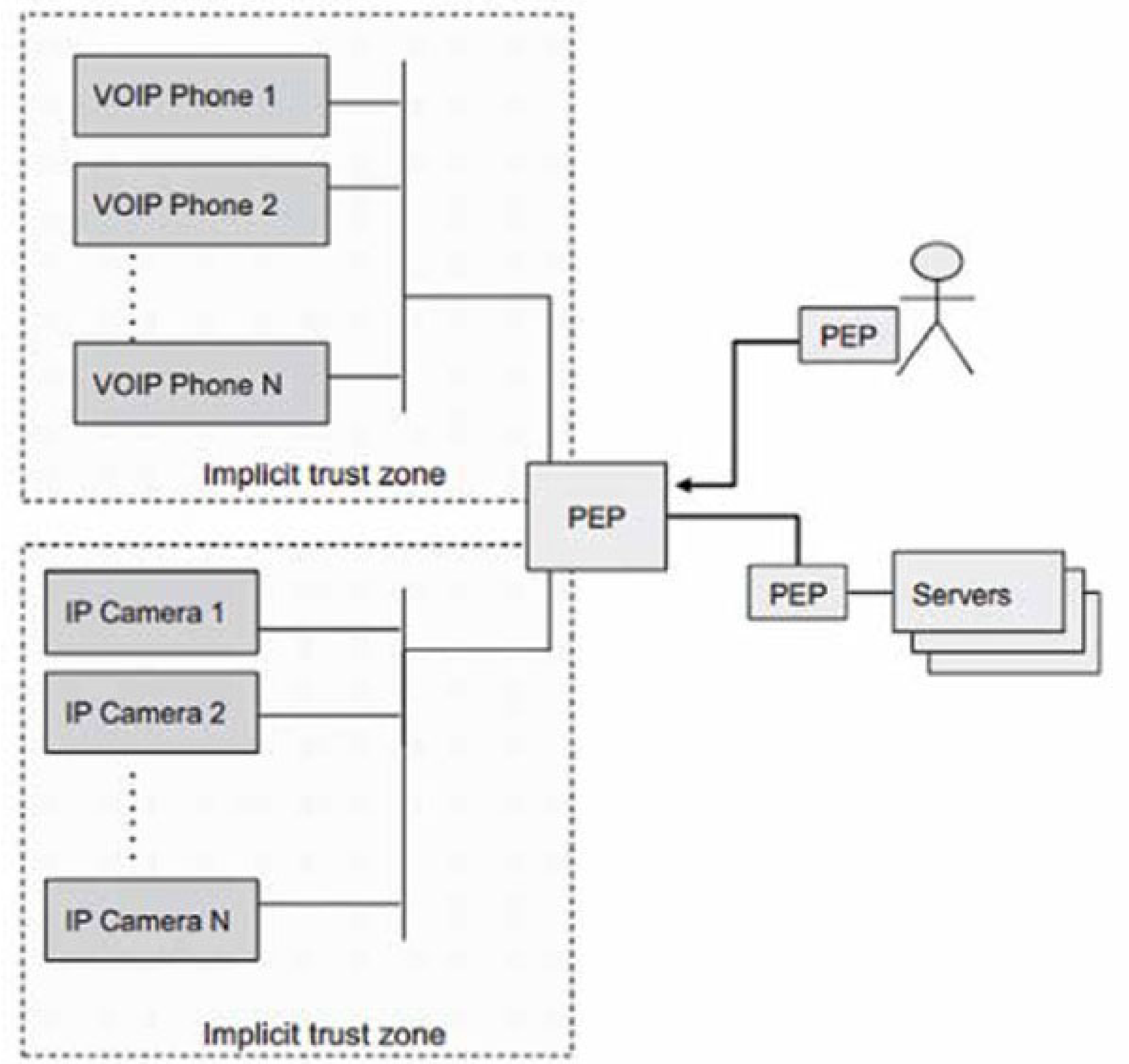


Figure 16.2: *Idealized Zero Trust IoT Network Model*

The advantages should be obvious, as this approach accomplishes each of the three objectives described above. To begin, the concept of least privilege is implemented: the PEP controls all upstream network traffic from each group of devices, implying that Zero Trust restrictions are imposed and enforced on outgoing traffic. This prevents data exfiltration, reconnaissance, or DDoS assaults from being carried out on a hacked device. Second, these devices are isolated inside their own uniform implicit trust zone, requiring inbound traffic to pass through the PEP, which implements access control restrictions in front of the open listening ports of the devices. Finally, all traffic between the PEPs is encrypted, bypassing any devices' native cleartext protocols. MITM attacks are much reduced as a result of this. Of course, even this idealized model is flawed, as these technologies are by their very nature. For example, devices within each implicit trust zone can interact directly over the LAN, thus if one IP camera is infected, malware may be able to spread laterally amongst peer cameras, however it will be limited to this isolated zone, with outgoing traffic restricted by the PEP. Another example: because device identification is frequently reliant on poor authentication procedures, an attacker may impersonate an IP Camera and get access to the same network rights as its peers. There are techniques to compensate for these flaws, which we'll discuss further below. Of fact, the idealistic perspective depicted in Figure 16-2 is a conceptual one, and real-world networks include a range of technological tools for identifying and authenticating devices, assigning networks and IP addresses, and routing traffic. These are the essential features of every network and security architecture, and they may be integrated in a variety of ways. Finally, a Zero Trust system for protecting IoT devices must be able to:

- Capture, route, and encrypt communications to and from these devices
- Manage access policies from a central location
- Ensure that access constraints are enforced throughout a dispersed network of devices

On flat, heterogeneous networks like the one shown in *figure 16.1*, this is difficult to execute in a reliable manner. The *tables 16.1* through *table 16.3* exhibit several methods to these functions, as well as their benefits and drawbacks:

	Pros	**Cons**
Physical Cable/ Switch Port	It's possible that the network is physically separated.	Change is difficult. The capacity of switch ports may hinder isolation. Isolating peer network devices is difficult.
Private VLAN	Within a single physical network, there is logical separation.	Physical network or switch port access is used to give access.
Wireless Access Point	Reconfiguring networks is usually easier. Many Wi-Fi systems include built-in device isolation.	Wi-Fi isn't available on all devices. Simple password authentication is insecure, and WPA-Enterprise isn't supported by all devices.

NAC/802.1x	Devices can be isolated by type via dynamic VLAN assignment. Frequently necessitates the purchase of pricey hardware.	802.1x is not supported by all devices. It’s difficult to keep track of a big number of VLANs.

Table 16.1: *How Devices Are Assigned to Networks*

	Pros	Cons
IP address	Fixed IP addresses can be used to identify a device.	Overhead in terms of configuration and administration. It’s a flimsy type of identification that’s easy to forge.
MAC Address	All devices are supported. On mixed networks, this is useful for identifying device classes and assigning them to zones (often with 802.1x).	Weak type of identity that is easily spoofable.
DHCP Fingerprint	Almost all devices are supported. On a mixed network, it’s a good way to identify device classes.	Weak type of identity that is easily spoofable.
Certificate via 802.1x	Solid and dependable.	Overhead for management and PKI. Many devices are unable to authenticate or identify themselves via certificates.

Table 16.2: *How Devices Are Identified/Authenticated*

	Pros	Cons
Default Network Gateway	Automatically assigned using DHCP. The natural policy enforcement point is a fixed and centralized egress point from the LAN.	On mixed networks, DHCP assignment may not always be able to distinguish between device types. Separate configuration from DHCP is feasible, however it may be time consuming.
Network router configures the path for protected resources.	Setup is simple and unaffected by device settings.	Access to destination resources is protected, but there is no way to filter by source. Access to other resources is not restricted on the device.
Manually configure devices	Route management at a finer level.	It’s time-consuming and tough to keep up with network changes.

Table 16.3: *How Network Routing Is Assigned*

The *figure 16.3* shows a collection of IP cameras put on an isolated and homogenous network, which is the simplest and most straightforward technique. A physically isolated wired network, a VLAN established by a NAC, or even a camera-only wireless network with an isolated SSID are all possibilities. What's key (and what makes it easy) is that it's homogeneous —all devices on that network are of the same type, and as a result, they all have the same set of network access rules.:

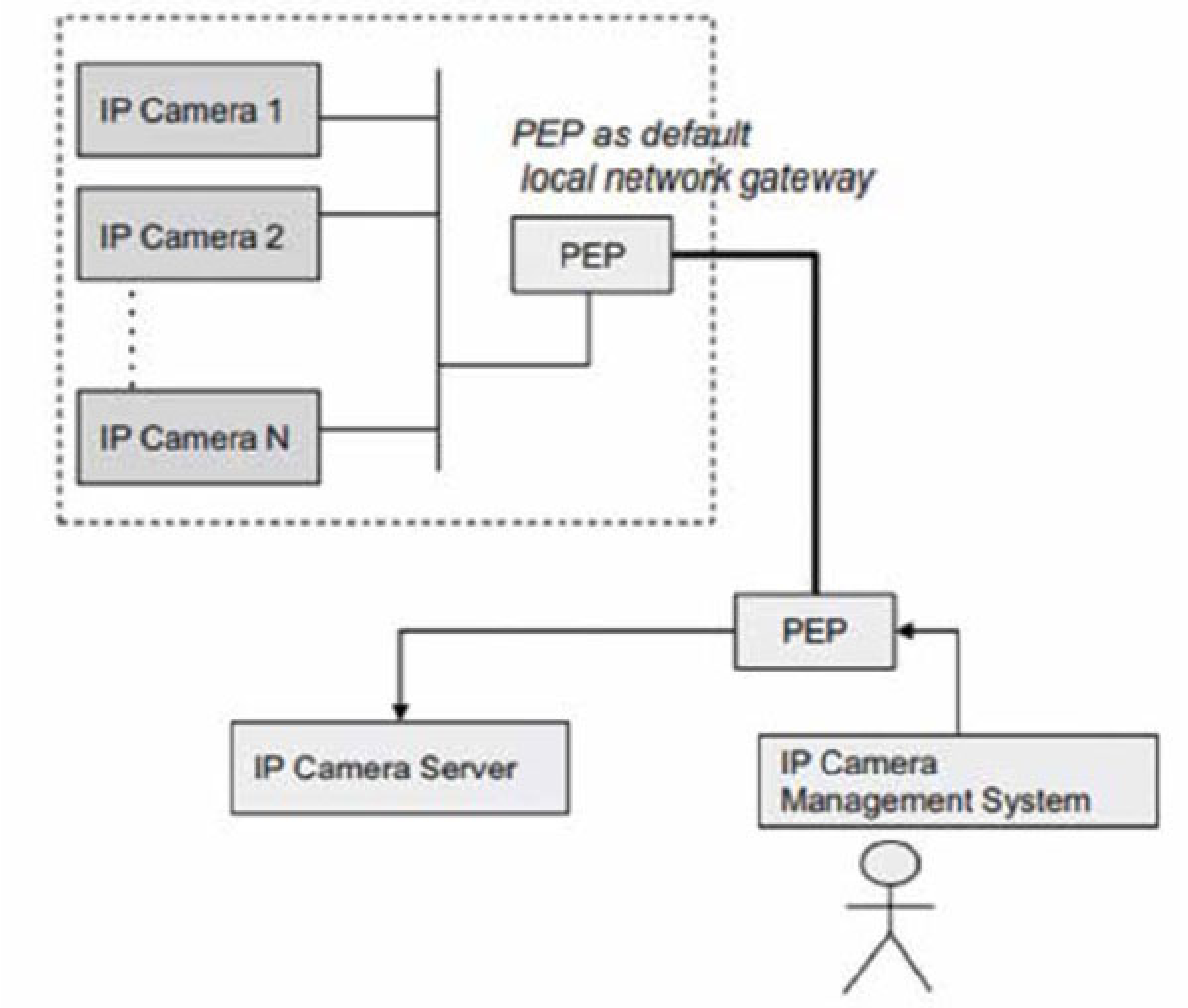


Figure 16.3: *IP Cameras on Isolated, Homogeneous Network*

The PEP is set as the default network gateway for the IP cameras on the network, which means that all non-LAN traffic is directed through the PEP, which conducts routing and policy enforcement. In other words, the PEP is the only way out of the local zone. For the camera-only section, the default network gateway assignment might be done centrally via the IP camera management system or via a DHCP server.

In any event, it should be obvious that this is as near to the ideal condition as feasible, making integration into a Zero Trust paradigm straightforward. The *figure 16.4* depicts a more typical scenario that is more difficult to regulate:

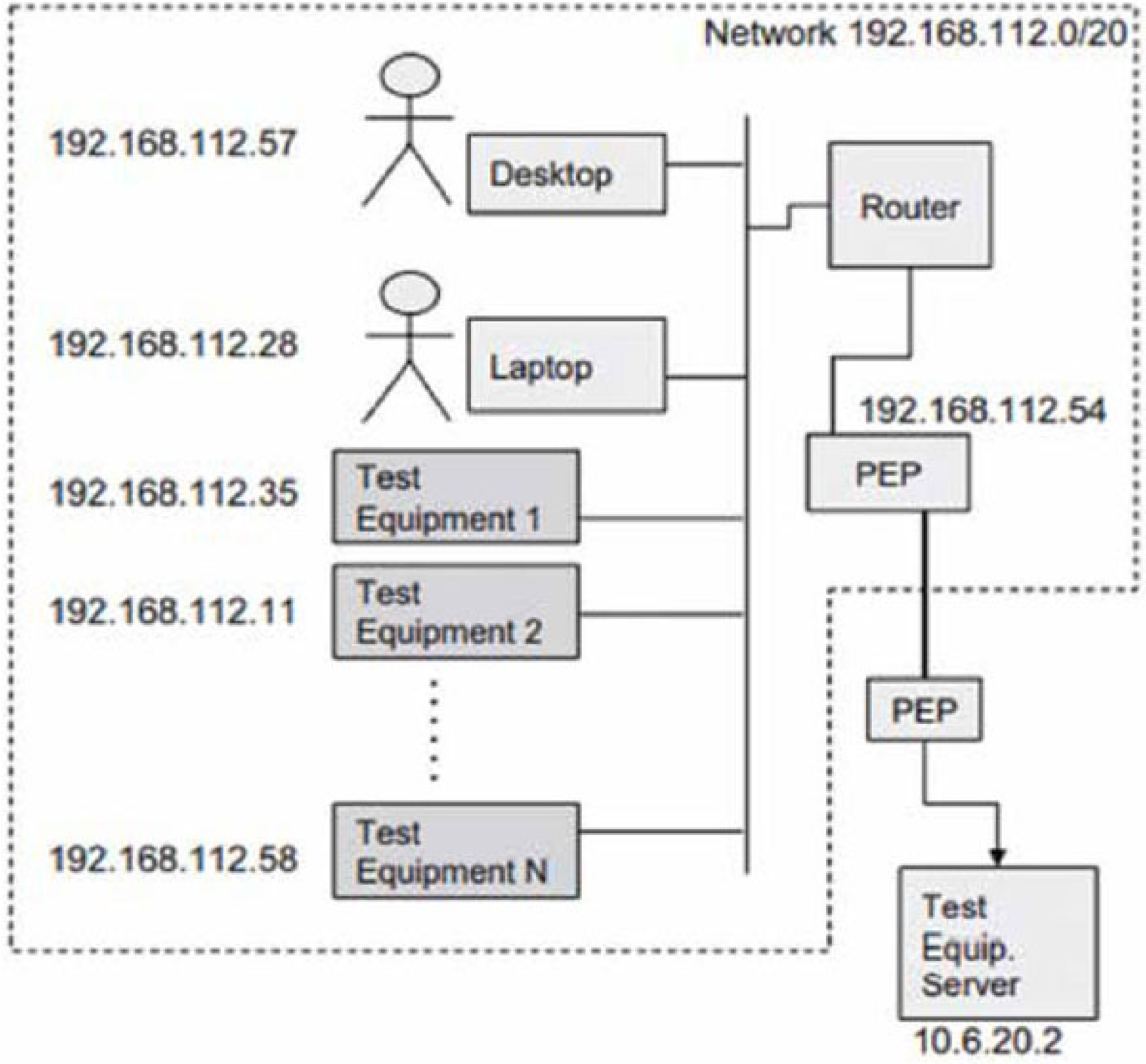


Figure 16.4: *Heterogeneous Enterprise Network*

This diagram depicts **192.168.112.0/20**, a mixed (heterogeneous) network segment made up of hundreds of computers and devices spread throughout a flat business network in a single office building. The devices on this network have DHCP-assigned IP addresses that are effectively random from the subnet range, and the organization lacks an accurate CMDB. This scenario presents a number of obstacles to achieving our objectives, which include ensuring that only Test Equipment devices are allowed to access the test equipment server, that no other devices are allowed to access the server, and that access to the test equipment devices is governed by policy. Unfortunately, in this real-world scenario, we won't be able to achieve all of these objectives without making network improvements. This is to be anticipated in many workplace circumstances, even outside of IoT devices—but at the very least, you should be aware of your network's flaws and figure out how to fix them, even if you can't do so right now. Let's look at what can (and can't) be done in this case, with an emphasis on protecting upstream network access from the test equipment. That is, the organization need a method to verify that traffic initiated by test equipment and sent to the remote test equipment server (**10.6.20.2**) is routed to the local PEP for enforcement and forwarding over the secure tunnel. This may be accomplished in three ways:

- Directly configured default gateway on Test Equipment
- Directly configured default gateway on Test Equipment
- DHCP-assigned default gateway on Test Equipment
- Static or dynamic network routing

It may be possible to set the default network gateway directly on the Test Equipment, depending on whether it is technically viable. It also depends on how time-consuming the procedure is. It's simple to do this with a centralized management system but needing individual manual modifications across hundreds of devices may be impossible. In some circumstances, using DHCP to set the PEP as the default network gateway for all devices may be a reasonable option.

If the DHCP assignment can properly distinguish between DHCP queries launched by the Test Equipment vs. those issued by other devices and return different results, the DHCP server may be able to provide various default network gates for distinct devices in some instances.

Finally, the network router may be configured to forward network traffic destined for the distant Test Equipment Server (**10.6.20.2**) to the local PEP (**192.168.112.54**). This has the advantage of requiring no further network modifications, but it does necessitate that the PEP be able to discern valid traffic (originating from a Test Equipment device) from fraudulent traffic (originating from malware on a user's device doing network reconnaissance).

Because IP addresses are issued at random and there is no CMDB in our circumstance, this will be tough to implement. It is conceivable for the PEP to utilize MAC addresses to differentiate devices, however they are a poor form of identification that may be easily faked, thus using them in this way entails considerable risk.

Conclusion

IoT devices, like many other aspects of our real-world networks and systems, are frequently complicated and difficult to maintain. In many (if not most) circumstances, Zero Trust can assist, but it often cannot provide the same level of protection as regular corporate devices (user systems and servers). Zero Trust PEPs may be used to limit upstream device access to protected resources, assure encrypted network traffic, and regulate downstream access to IoT devices—all with variable degrees of efficacy, depending on the numerous criteria we've explored in this chapter. There are a few qualities you may look for to assist you discover well-suited IoT systems when you look at your company to find prospective IoT systems for inclusion in your Zero Trust project. To begin, learn how these devices' networking is set up, and look for IoT devices that have a centrally managed method for scalability. Second, seek for sections of your network that are well-documented and well-understood. As an early project, avoid attempting to protect IoT devices on an unmanaged, heterogeneous, and opaque network—you should have some expertise and success applying your Zero Trust architecture to IoT systems in simpler and more well-understood contexts first. Finally, search for "low hanging fruit" in terms of protecting third-party remote access to internal equipment. The ability of Zero Trust to demand a business activity prior to access, such as the production of a service desk ticket, may immediately bring meaningful security benefit. IoT devices are a relatively new area for Zero Trust, and as we've seen, they're frequently sophisticated and technical, as well as a minefield of outdated, inflexible technologies. However, there is a lot of room for development, and we've just scratched the surface on this subject—it could be a whole book in and of itself.

CHAPTER 17

A Policy of Zero Trust

Zero Trust is all about policies—PDP and PEP are the two main architectural components. Of fact, the term policy is significantly overloaded in the English language, with several connotations. Policies are the frameworks built by organizations in our Zero Trust environment to determine which identities are allowed access to which resources and under what conditions. Remember that in a Zero Trust environment, access is only granted once a policy has been evaluated and assigned to an identity, and that access can be enforced at the network or application level. The actual, technical methods for defining and enforcing policies will vary by product and implementation, but the principles and components are universal and should be present in every Zero Trust system. "Policy is the set of access rules based on attributes that an organization assigns to a subject, data asset, or application," according to the NIST Zero Trust document, and *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*, according to one of the NIST core tenets. Remember that Zero Trust must "allow the dynamic enforcement of security policies," as we defined it in Chapter 2*: Get to Know Zero Trust*. We're bringing structure and detail to the policy discussion, extending some of the principles under the NIST Zero Trust framework, as we briefly mentioned in Chapter 3*: Architectures With Zero Trust*. We're also incorporating and exploiting industry technologies such as ABAC, which we're reinterpreting from a Zero Trust viewpoint. The purpose of this chapter is to give you with a framework and structure for thinking about the breadth of your Zero Trust system and evaluating vendor platforms. Because the breadth and depth of Zero Trust systems may be practically endless, it's critical that you have a clear understanding of what should and shouldn't be included, as well as what's appropriate to include in a policy model. This is essential before you can start defining your policy architecture and lifecycle, as well as the governance mechanisms that go with it. Understanding the capabilities and limitations of your proposed policy model may also help you create criteria and constraints for your Zero Trust architecture. Let's get started by going through the logical components that make up policies in detail.

Components of policy

We revisit the policy framework from Chapter 3: *Architectures With Zero Trust* in this part, this time with extra discussion. The policy structure is depicted in *table 17.1*, which defines the subject criteria, action, goal, and condition components:

Components	**Description**
Subject	The beings executing (initiating) acts are called subjects.
Criteria	Subjects must be verified individuals, and policies must provide subject criteria that identify the people to whom the policy applies.
Action	The activity that the person is engaged in. Either a network or an application component is required, and both may be present.
Target	The activity that the person is engaged in. Either a network or an application component is required, and both may be present.
Condition	The conditions in which the subject is allowed to carry out the action on the target. The Zero Trust system must allow for the creation of conditions based on a variety of factors, such as subject, environment, and target properties.

Table 17.1: *Zero Trust Policy Model Components*

It's important to note that what we're showing here is a logical framework, which we feel is a good approach to think about policy components. Actual Zero Trust implementations may have a different policy model structure, but they should have these features. Let's take a look at each component of the policy model one-by-one.

Subject criteria

In the end, the prescribed action will be performed on the target by a subject (an authenticated identity). The PDP assigns policies to subjects and analyses each policy's criteria against the subject in question at various times (we'll go over this in more detail later in this chapter). Note that policies don't usually relate to a specific subject; instead, they include the criteria that the PDP employs to determine if a policy is assigned to (applicable to) a certain identity. The requirements for some rules might be wide ("*All workers*") or specific ("*Users in group Marketing, assigned to project Bruin, and utilizing Windows devices*"). Directory group membership, identity-assigned attributes, and relatively static device attributes like Operating System version and patch level, or mobile device jailbreak status, are all common topic criteria. It's worth noting that, as we've explored throughout the book, subjects don't have to be human users. Servers (or, more logically, the service accounts that run on them) can have identities and so be authenticated subjects with policies assigned to them that enable them certain access permissions. Note that the technique we're presenting here is the criteria-based approach through which the PDP's trust algorithm generates policy assignment choices, as described in the NIST article. A score-based technique is also discussed by NIST, which is acceptable. We're not favoring one over the other, but for the sake of this debate, we feel it's easier to think of a set of criteria that must all be met in order for a policy to be allocated to a certain issue.

Finally, keep in mind that we're looking at policies and subjects from the standpoint of subject-initiated activities, which originate from the well-known situation in which a user or a server (both authorized subjects) connects to a server over a PEP to access a resource. Later, we'll look at a more complicated example in which this link is made in the opposite direction.

The policy's actions describe the kind of activities that are authorized. While many of the actions will be connected to network access, it's also probable that certain Zero Trust systems will include the capacity to enforce other sorts of activities, such as application or data-centric actions. This distinction refers to the various sorts of PEPs that function at the network and application levels, respectively. Actions should indicate the permissible set of network ports and protocols from a network perspective. Actions might be linked to roles, characteristics, application services, or data categorization in an application or data (more on this topic shortly).

We feel that thinking of actions as being specified independently of the targets for which they will be carried out simplifies things, while some implementations may mix the two.

Here are some actions to consider:

- Use HTTPS (TCP on port 443 with TLS) to access the resource
- Use HTTPS (TCP on port 443 with TLS) to access the resource
- Use TCP on port 3389 (RDP) to access the resource
- Use UDP on port 53 (DNS) to access the resource and receive a response (Windows SMB)
- Perform a Linux kill command through SSH
- Access data marked as "unclassified" with read/write rights
- Access data tagged as "customer PII" with read-only permissions
- Access data tagged as "unclassified" with read-only permissions

Our examples cover TCP and UDP access (regulated by a network-level PEP), as well as a few that rely on application-level principles (and application-level PEP enforcement). These latter capabilities are intriguing, and they are unquestionably more "*leading edge*." Currently, Zero Trust regulations do not include application-level operations (application functions). This is due to the fact that most apps use an opaque, internal authorization mechanism that cannot be managed by an external system. However, with the emergence of contemporary HTTP-accessible online apps, we're now seeing a more regular link between application functions and URLs, allowing for PEP-enforced actions in ways that weren't before conceivable.

The data samples, as well as possible actions on other apps, are more forward-looking notions. The notion is that a Zero Trust system would employ an open security architecture that would let the PEP and the application to share information that would help them enforce policies more effectively. For example, we may imagine an application that connects to a PEP via a plug-in or configuration and correlates application protocol components with application activities, allowing the PEP to impose rules or even execute Just-In-Time application role creation for the subject. Alternatively, a systematic mechanism for the PEP to transmit extra identification or contextual information to the application, allowing the application to apply Zero Trust rules. You may recall that Google used this strategy in their BeyondCorp campaign, which we discussed in *Chapter 4: Zero Trust in Practice*. Their Access Proxy (essentially, their PEP) inserted extra contextual information into HTTP headers for user activities, which the target apps may either ignore or use. We believe this is a developing field with exciting advances expected in the next years. We'd want to see an open platform that allows application developers and Zero Trust systems to communicate and collaborate. Even if your business doesn't have this technological interface, keep in mind that access governance rules should be in place to guarantee that users only have the right application roles and capabilities. This is an excellent illustration of how different types of systems and different portions of an organization may function in harmony.

Target

The host, system, or component that will be operated on is defined by the target. Targets can be defined statically or dynamically (requiring action from the PEP to completely render the target). One of the major ideas of Zero Trust, and one of the reasons why it's so fascinating, is dynamic policies. These rules enable you to establish and enforce access based on unknown and unknowable properties until runtime. Let's have a look at several target samples that show a variety of various sorts.

10.6.1.34 is the IP address of the host

The network PEP does not need to perform any further effort to ensure this because it is a basic, static, and completely rendered target. Targets that specify a single IP address, while helpful, are rarely a viable choice for inclusion in a policy. Of course, IP addresses vary, and in many circumstances, the logically intended access is to the application or service operating on that IP address rather to the host at that IP address. In such circumstances, a hostname rather than an IP address may be a preferable target. In the next example, we'll talk about this. Finally, there are times when specifying fixed IP addresses within targets makes sense, especially if access is being allowed to IT or network managers who require access to infrastructure elements such as networking gear. You are the most knowledgeable about your surroundings, and you will need to choose the most successful technique in the end.

Access to Host appserver1.internal.example.com

The usage of targets that specify hostnames is fairly popular, and it’s a great approach to build rules. DNS, of course, is used to map hostnames to IP addresses. Policies using a single hostname as a target allow you to construct fine-grained access controls, and they’re typically paired with a small number of actions, most commonly a single network protocol and port.

Zero Trust policies must be able to use and operate with an organization’s internal DNS systems, teams, and procedures, rather than against them. Consider an internal user-facing programme whose IP address changes on a regular basis.

For a virtualized system with a rolling set of application updates that takes a staggered approach to production rollouts, this is absolutely fair. DNS is also commonly used for load balancing, geographic distribution among a group of replicated application servers, and simply as a general best practice to allow IT teams to make network modifications independent of applications.

It’s critical for the Zero Trust system to be able to resolve hosts by having distributed PEPs utilize the right DNS server in all circumstances. Because all hostnames are in distinct domains and/or on unconnected networks in a distributed Zero Trust system running over many distant networks, a centralized PDP is frequently unable to resolve all of them.

Access to hosts on the subnet 10.5.1.0/24

This example displays a static target that corresponds to many hosts inside the subnet and, in reality, enables access to all hosts on that subnet. This kind of openness isn’t ideal, and it usually goes against the notion of least privilege. However, there are always exceptions. This target may be useful in a policy if it were given to IT administrators with a valid need to access all hosts on the network, for example. Alternatively, if the network was split such that all of the devices on it were of the same sort, it made sense to provide access to all of them. This target is most likely to be utilized in a transitional condition, when an organization is on its way to Zero Trust and isn't ready to implement finer-grained access restrictions. With the implicit trust zone behind the PEP, this might be implemented as an enclave-based approach.

Access to systems tagged as "department=Marketing"

Because it depends on the PEP to actually resolve the hosts after the policy has been issued to the subject by the PDP, this example demonstrates some of the true potential of a Zero Trust system. This flow will be discussed more in this chapter, but for now, consider how the Zero Trust system employs the PEP to completely render the policy based on its capacity to probe its surroundings. That is, the policy author is depending on the organization's usage of metadata inside the runtime system to specify the workload's contents, which defines who may access it. The intricacies of how a "*tag*" (also known as a label in certain systems) is applied and resolved will vary depending on the implementation, but they are not important here. What matters is the concept—those enterprises may manage access using a method outside of normal IT and networking, which, for the first time, links together business or technological operations and security. For example, an organization may get this information via a property in their **Configuration Management Database** (**CMDB**) or a metadata attribute like a tag in an IaaS environment. In both circumstances, the Zero Trust consumption of this metadata serves as a strong and automatic integration point, allowing IT and security teams to communicate. Any host marked in this fashion would be immediately identified by the Zero Trust system and given the proper set of access regulations, implying that the relevant set of users would be given the appropriate degree of access merely by using this tag. It's also worth noting that this example expands the policy model to include forms of resources that aren't purely host-based. The majority of today's current apps are containerized and/or microservices based and are not directly coupled to their underlying host. Regardless of whether many services are running on the same physical or virtual host or share a shared IP address, the Zero Trust policy model must be able to distinguish between them and impose various degrees of access. Service mesh systems like Istio, for example, have a policy model that matches the approach we describe here. The service mesh is made up of a distributed group of PEPs with authorization rules that contain a tag-based system for selecting policy targets and a mechanism for specifying conditions.

Access to systems tagged as “stage=test”

This is identical to our last example, but with one key difference: the usage of a deployment stage tag. When combined with an automated toolchain that releases new versions of apps or services in a continuous basis, the ramifications are huge, especially in a DevOps context. In this case, the toolchain would utilize the tag to signify the proper development lifecycle stage, and the Zero Trust system would provide access to these targets to the appropriate group of subjects (human or system). This implies that as a result of the deployment operations, subjects will gain the appropriate access automatically and transparently. When the stage of a workload or service changes, the access controls for that workload or service change as well. This, in our opinion, perfectly demonstrates the strength of a Zero Trust system. It takes use of previously completed technical effort (toolchain-driven deployment) by employing an attribute to automatically alter access rights. As a result, as workloads proceed through their lifecycles, they keep exactly the proper set of basic access constraints without requiring any user intervention. This technique may be easily integrated into a DevOps company, allowing them to maintain their speed while adhering to Zero Trust security standards.

Condition

The conditions define the circumstances under which the subject is permitted to carry out the action on the target. Note that policy models should be able to check for a wide range of situations; in particular, your Zero Trust implementation should be able to provide an expandable set of criteria so that custom checks may be added. Although certain Zero Trust implementations may allow more kinds, conditions are often checked against device, authentication, or system-level properties. Let’s look at a couple of examples of requirements that must be met in order for access to be granted.

Between 08:00 and 18:00 is the time of day.

Users with well-defined jobs and regular hours benefit from time-of-day limits, which are a practical and concentrated method of access management. This condition protects against stolen credentials and malware, both of which might try to access resources outside of business hours. This condition is also beneficial for always-on device scheduled maintenance periods; consider a group of retail devices that need to connect to an IT back end every night between 01:00 and 03:00. There's no need to allow that network connection unless it's for that specific time period. This situation exemplifies why the PEP must be able to render the policy in its entirety. The PEP must be able to compare the current time with permissible time windows throughout the day, even if an identity only authenticates with the PDP once per day.

User has performed a valid MFA or step-up authentication within the last 90 minutes

Within the last 90 minutes, the user has completed a valid MFA or Step-Up Authentication. We believe that MFA should be used appropriately, and that this form of condition should be a required (and common) component of every Zero Trust implementation. Of course, deciding when and how to demand step-up authentication is a balancing act that must consider both the user experience and the threat model you're fighting against. Certain high-risk or high-value apps may justify requesting for MFA every time a user starts a session with them, but in many circumstances, it's just as effective and less invasive to demand MFA once for a collection of resources and have that valid for a period of time. Again, the PEP must evaluate and enforce this sort of condition; step-up authentication might be triggered at any moment depending on user access through the PEP, and the PDP is unlikely to be engaged in that flow. As a second factor, there are several ways and solutions available, including FIDO2, smartphone apps, push notifications, and biometrics. Any or all of these should be supported by Zero Trust systems via standardized APIs, allowing you to pick and choose which ones work best for your context.

Anti-malware service is running: device posture meets requirements

This condition verifies that the subject’s device complies with security posture standards. It’s worth noting that this example relies on data collected from the device itself. The user agent PEP or another software component on the device can verify whether the anti-malware service is currently running, but bear in mind that any information supplied from a device should only be regarded partially reliable. While many IT and security firms practise good security hygiene on user devices, such as limiting administrative capabilities, malware on the device might still be producing misleading data.

While knowing that a device’s anti-malware service is active is useful information and an acceptable condition to impose, it should only be regarded one part of a defense-in-depth strategy.

Endpoint security scan completed in less than 48 hours: device posture meets requirements

This condition utilizes data from a security scanning tool, such as an endpoint management or vulnerability scanning solution, to determine device posture. Because the PEP obtains this information from a server rather than a device, it can be considered more reliable. It’s worth noting that in this case, the condition necessitates the completion of a recent security scan. Another option is to use more up-to-date information, such as having the PEP call a monitoring system like a SIEM or UEBA and acquire near-real-time information about the risk level of a certain device.

For this resource, a service desk ticket is now open.

This is one of the most fascinating and convincing instances of how Zero Trust systems may connect security enforcement and business operations together. This allows enterprises to use their preferred business processes, similar to the metadata tag example we mentioned for Targets. It ensures that users will follow the process by making access—enforced by the network or application—a result of a successfully conducted business process. This can provide significant advantages in terms of auditability, repeatability, and quality, as well as security.

In this case, the business wishes to make sure that IT admin access to a certain resource is only allowed (and possible) if there is an active Service

Desk ticket for that resource. As a result of this policy, stakeholders will be needed to submit a Service Desk ticket in order for an IT administrator to get access to the resource and complete the assignment. Admin access to that resource is withdrawn after the ticket is closed. This reduces the requirement for admins and their devices to have constant network access while yet allowing them to be completely productive. It also guarantees that every admin access is monitored in order to assure compliance.

***This Service Desk** ticket is just one example; Zero Trust technologies may be linked with virtually any business activity in the same way, resulting in significant organizational advantages.*

The subject and the target must both be servers in the "production" state.

The subject and the targets in this example are both servers, and their states are indicated by a tag. This condition prevents development or test apps (or developers working on non-production systems) from connecting to a production service accidently. Of course, depending on the service-to-service authentication architecture, further levels of restrictions via authentication may be required, such as application credentials or a certificate. However, it's all too easy for developers to make a mistake, especially in settings with manual testing or release stages; typically, this work is done via the command line, where a single copy-and-paste or typographical error may have a huge impact. Further scope limits, such as by application or project name, might be added to this form of control, which uses both subject and target service metadata.

While it's improbable that an application service from project oriole would connect to a service from project blue jay by accident, it's feasible that a malicious person or virus with access to the application's host may try network reconnaissance or lateral movement. Once our Zero Trust architecture and security maturity are ready for it, our concept of least privilege should push us to have these sorts of regulations in place.

Conditions vs. subject criteria

As we've gone through these examples, you've probably realized that several of the checks may be classified as topic criteria or conditions. That's fine—there aren't any hard and fast rules here, and you'll have to make some educated guesses based on your organization's needs and the platform you've selected. As you acquire expertise, you should be able to tell which strategy is the most effective. Consider that some sorts of checks, even though the PEP is technically capable of executing them, will be more suitable to conduct in the PDP at initial session setup (such as during identity authentication). These tests are usually for properties that are slow to change and, as a result, are likely to stay the same for the duration of a user session. Of course, it depends on how the Zero Trust platform is built, but OS version and geolocation, for example, are unlikely to change while a session is running. Later in this chapter, we'll look at qualities and where they should be assessed.

Policy examples

Let's put them together in a few example policies to show some of the ways these components might be weaved together now that we've looked at how policies are created and examined some instances of their components. Our first example policy is the one displayed in *table 17.2*, which we presented in *Chapter 3*: *Architectures With Zero Trust* when we initially described the policy model:

Billing department users must be able to access the Billing web application.	
Subject criteria	Users who are members of the Identity Provider's Dept Billing group
Action	Users must be able to access the Web UI on port 443 over HTTPS.
Target	The billing application with the FQDN billing.internal.company.com.
Condition	On-premises or remote users are also possible. Remote users must be asked for MFA either when they first log in (at the time of authentication) or once every four hours. Users must access this application through a company-managed device that is protected by endpoint security software.

Table 17.2: *Sample Policy—User Access to Billing Application*

The topic criteria in this case will apply this policy to users who are members of the Dept Billing identity provider group. Because only employees are kept in this organization’s identity provider, there’s no need to include a check for that position in the criteria because it’s implicit. It’s also worth noting that they didn’t include a check to see if the user had an active account in the billing application in this example. This may have been handy if certain employees of the Billing department, but not all, are regular users of the application. The topic criteria in this case will apply this policy to users who are members of the Dept Billing identity provider group. Because only employees are kept in this organization’s identity provider, there’s no need to include a check for that position in the criteria because it’s implicit. It’s also worth noting that they didn’t include a check to see if the user had an active account in the billing application in this example. This may have been handy if certain employees of the Billing department, but not all, are regular users of the application. This is an intriguing topic, because it exemplifies a very regular situation in which an organization’s identification group does not completely map to the set of users who should receive a certain action. Of course, the ideal case for Zero Trust rules is to apply the principle of least privilege and only allow access to the exact set of individuals that need it. However, we believe that moving forward with a policy that grants access to a few extra users is preferable to waiting for changes to an identity management programme and process to achieve a “perfect” group mapping, and that it is better to move forward with an imperfect Zero Trust implementation than none at all. Remember how we spoke about this in *Chapter 5: Identity and Access Management (IAM)*? This is an excellent illustration of how a Zero Trust project can and should carry forward and create value even if the identity team is working on something else.

Returning to our example, the action is straightforward—just the ability to access the web UI over HTTPS—and the target is a straightforward fully qualified domain name. The conditions, on the other hand, are more intriguing since they are utilized to impose a few restrictions. Remote users must first submit an MFA prompt while accessing this programme, and then again four hours later.

Because their actual presence in the building might be regarded an extra element, users operating on the (more) trustworthy internal corporate network are not required to submit MFA. In addition, the device must be administered by the company (as evidenced by the existence of a valid certificate issued by the company CA) and run the company's endpoint security solution. In our opinion, this is a sensible and balanced set of criteria, allowing remote people to remain productive while inflicting little irritation and requiring access only via genuine company-managed devices. Let's look at another scenario, which is based on certain limits in their surroundings and is in some ways even simpler:

Policy: Sysadmin access to production subnet	
Subject Criteria	Users who are members of the identity provider's Sysadmins group
Action	Users can use ICMP ping to reach TCP ports 22, 3389, and 443.
Target	Any host on the 10.0.0.0/8 subnet.
Condition	There must be an open service desk ticket that identifies the hostname or IP address that is being accessed.

Table 17.3: *Sample Policy—Admin Access to Production Subnet*

The policy is being used to limit administrator access to production servers in the example in *table 17.3*. Their sysadmins require remote access to servers or network devices in their huge production subnet, which encompasses hundreds of hosts (SSH, SFTP, Web, and RDP). These sysadmins must connect to a couple of those systems on a daily basis to update, change, or troubleshoot. The company doesn't want their administrators to have constant access to this network, but they do need them to be able to conduct their tasks, which means they require access to an arbitrary and unexpected collection of hosts on a daily basis.

By connecting access control to a business process—the usage of their service desk (ticketing) system—this sample policy solves their problem. This strategy allows the company to keep its administrators working while also assuring that all access to the production systems is recorded.

It's worth noting that, like many real-world businesses, our hypothetical company is more mature in certain areas and less mature in others. Their usage of the service desk as a dependable and consistent mechanism for conducting sysadmin jobs demonstrates a high level of maturity in this case. On the other side, the fact that each server has both SSH and RDP access implies that they don't have a reliable asset management system to match hosts to OS types.

There is also no MFA linked with this particular policy. Perhaps there is a cultural aversion to utilizing it in this fictitious organization, or perhaps the organization uses a compensating measure, such as a credential vault. The sample policy in table 17.4 shows how to use a dynamically generated target in an IaaS context, with the PEP evaluating the metadata for targets. While the activities are wide open, considering that this is a development environment, this is reasonable. This policy allows developers to operate freely within their IaaS environment while preventing them from accessing the resources of other projects:

Policy: Developers accessing project "Everest" resources	
Subject criteria	The topic must be a member of the directory group Project Everest.
Action	Actions on TCP, UDP, and ICMP
Target	Any resource labeled with "project=Everest" in the IaaS development environment.
Condition	None at all (access is always permitted).

Table 17.4. *Sample Policy—Developer Access*

Policy: Web server in DMZ accessing database server	
Subject criteria	The hostname of the topic must be ws1.company.com or ws2.company.com.
Action	TCP port 3306 may be accessed.
Target	App1database.internal.company.com is your host.
Condition	App1database.internal.company.com is your host.

Table 17.5: *Sample Policy—Server-to-Server Access*

In table 17.5, the last example policy depicts a server-to-server situation, demonstrating how a business may design a policy allowing a web server in the DMZ to contact its related database server on the private internal network. This web application might easily serve as the front end for an e-commerce site, with a variety of back-end services interacting with the main database (for instance, updating inventory, or processing orders). There are numerous instances of the web server in this example for load balancing and high availability, and their internal IP addresses change on a regular basis owing to the underlying virtual architecture, with new versions being delivered often. This simple policy allows the business to automatically alter access while maintaining strong security, even as their infrastructure changes.

Applied policies

This policy model should give you a framework for thinking about and creating access rules that are understandable to both technical and non-technical stakeholders. It should also be useful when you evaluate possible Zero Trust products, since it will give you an idea of the skills you require. Those policies, of course, do not exist in a vacuum—in order to be reviewed, they require characteristics (contextual inputs), they must be built to match certain situations, and they have a flow connected with them in terms of when and why they are evaluated. Starting with characteristics, we'll look at each of these features of policies in this section.

Every business will very certainly build custom groups and give custom characteristics to its identities and using these attributes in policies in Zero Trust systems is a key, must-have skill. Device characteristics can be collected either directly from the device (through a local agent) or from an external system like a CMDB or an endpoint management system. Some device properties, particularly those received from a device's local process, are subject to frequent modification. We examine the relative durability of traits in the following section, and it should obviously be considered when deciding when, when, and how to assess them.

Another collection of qualities to think about is what we call system attributes.

This category is a bit of a catch-all for qualities related to the enterprise network and ecosystem as a whole. This might contain things like the overall network threat or risk level (perhaps collected from a SIEM), system or network load, or even attributes related to business processes or IT activities, such as if this is an allowed maintenance window or whether an emergency IT "*breakglass*" scenario exists. Finally, as previously mentioned, target qualities are utilized to define action targets. These might be obtained by having the PEP query its local environment, or from a centralized and authoritative source like a CMDB.

It's worth noting that we've discussed attributes from the standpoint of having the Zero Trust system retrieve attributes from external sources. While this is likely to be the most prevalent case, it is not the only one. It's quite logical for the Zero Trust system to serve as an attribute repository. Of course, a set of processes to populate and update the characteristics is required in this case—recall that we addressed this in *Chapter 11*: *Security Operations* when we explored security orchestration.

Let's look at the pace of change of various sorts of qualities in light of this.

The *table 17.6* contains a table of attribute permanence, as well as samples of attributes and their overall frequency of change. These should not be seen as hard and fast rules, but rather as a collection of suggestions. Even "permanent" biometric traits can alter owing to injury or transplant, for example. And your company's asset management policies, for example, may have an influence on the relative durability of particular device properties. Overall, we think you'll find this table beneficial since it may help you comprehend the different sorts of qualities and decide when and how often you should evaluate them (for example, at authentication time vs. at access time):

Attribute permanence	**Identity attributes (users)**	**Device attributes**	**System attributes**	**Target attributes**
Permanent (never change)	Biometrics (for example, fingerprints, iris scan)	Operating System	None	Operating System
Semipermanent (fewer than one change per year)	Citizenship Country of Residence Certifications Security Clearances	Hostname	Domain	Identifier Hostname URL
Infrequent (monthly or yearly changes)	Group memberships Roles Project Assignments	OS version or patch level Component patch level (for example, AV signature file)	DNS Server Settings	IP Address Certificate info Network info (for example, TLS parameters) Resource Version
Regular (weekly changes)	None	Device posture check Registry key values	None	Target posture check
Frequent (hourly or daily changes)	Geolocation Network attributes	Process Status Device IP address	Network risk level Network load Breakglass Situation	Resource Load Resource Availability

Table 17.6: *Attribute Permanence*

In reality, validating regularly changing features in the PEPs, most often as part of a condition, makes sense. This is due to the fact that these properties might change throughout an active session, and the PEPs are the mechanism for enforcing access-time restrictions. Longer-lasting characteristics might be evaluated as part of the PDP's subject criteria. Of course, keep in mind that your selected Zero Trust platform may take a different approach.

Now that we've looked at the structure of policies and the characteristics that are used as input, it's time to look at the most prevalent circumstances. This refers to the patterns and procedures through which subjects get access to targets. But first, let's double-check that we comprehend our standards and assumptions. First, each Zero Trust action must always include at least one subject (with the exception of IoT systems, which we explored in Chapter 16*: IoT Devices*). Subjects are verified identities. Unauthenticated identities, on the other hand, cannot be subjects but can surely be targets.

Second, there will very certainly be some amount of communication on a network outside of the Zero Trust system's control, but inside the implicit trust zone. The bounds of your implicit trust zone must be thought about and clearly decided upon, and it should diminish over time as you move on your Zero Trust path. During your trip, you'll encounter a mixed bag of communications with Zero Trust resources, with some communication going through PEPs (and so subject to Zero Trust policies), and other communication going through the Zero Trust PEP (and thus bypassing Zero Trust policies).

Finally, resources that are available to unauthorized users (such as public web servers) are not covered by Zero Trust policies. These systems purposefully provide any distant system a level of confidence, allowing it to connect to and use the resource. Other services, like as administrative access to the host, can (and probably should) be included in a Zero Trust system even on a publicly available web server. In other words, Zero Trust's demand that all subjects have validated identities will aid you in defining unambiguous system boundaries.

While certain resources in your environment will be beyond the purview of Zero Trust, they will undoubtedly stay within your security team's scope and must be secured with proper controls.

Now let's look at a few situations that combine our policy model with Zero Trust deployment approaches. For clarity, we have removed the PDP from all of these diagrams:

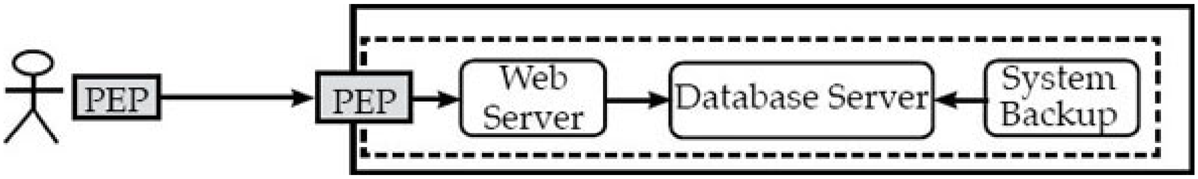


Figure 17.1: *Policy Scenario—Web Server as Target*

The *figure 17.1* demonstrates a straightforward deployment utilizing the enclave based Zero Trust deployment methodology. The web server is the single policy target in this example, and it controls the user's access to it. Because all three servers are in the Zero Trust implicit trust zone, the PEP has no control over their access to one another. That is, the PEP merely restricts the subject's access to the web server:

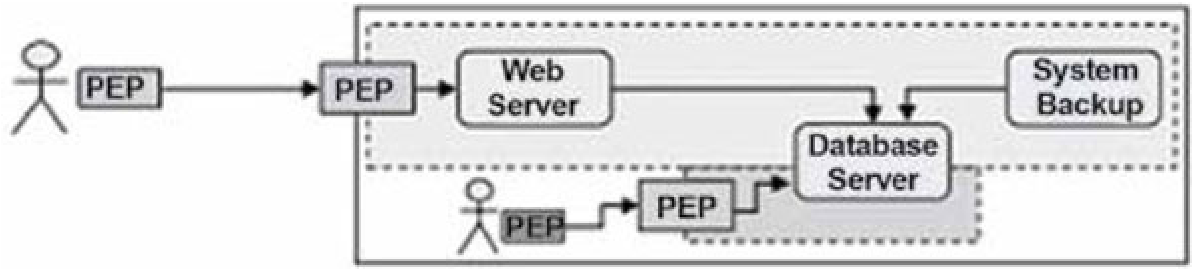


Figure 17.2: *Policy Scenario—Database Server as Target*

The database server has been partially hidden behind a PEP in the case depicted in figure 17.2, allowing it to be a target inside a policy. In this case, the business has limited IT administrator access to the database server, allowing only authorized identities to access it via an established policy. The web server and backup system, on the other hand, continue to connect directly to the database server since they are in the implicit trust zone. In reality, this would be accomplished with firewall configurations that limit admin access (through port 22) to the PEP while allowing database access (via port 3306) to any other server inside the broader implicit trust zone. This is a good real-world example of how an organization can gradually move to Zero Trust: by treating the various services running on the database server as separate logical targets, they can improve their security by strictly controlling admin access without affecting their application or business operations. The figure 17.3 represents a possible next stage in this trip; it may or may not be a brief phase:

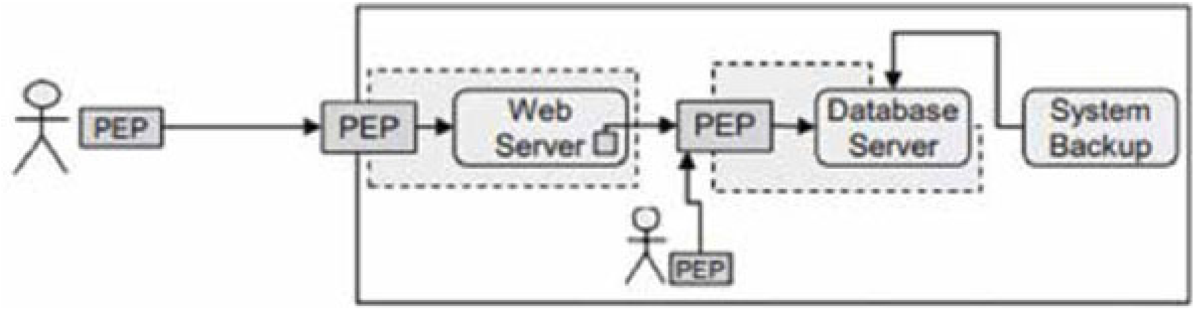


***Figure 17-3:** Policy Scenario—Web Server as Subject*

The database server has also been put behind a PEP, so that only authorized people can access it, as seen in *figure 17.3*. As a result, the web server must now be treated as a subject with an identity and a method of authentication. The end result is a smaller implicit trust zone (which improves security) and more deployment flexibility.

This later benefit of Zero Trust is an intriguing and sometimes neglected feature. Because the web server's access to the database server is now controlled by the Zero Trust system, the database may be moved anywhere in the enterprise's infrastructure without affecting the web server, with the exception of perhaps increased latency.

That is, the database server's deployment location is no longer significant to the web server, and it may be moved to a distant or cloud-based location without affecting the web server's performance. Without Zero Trust, the company would be unable to accomplish this without some form of remote access, often over a WAN link. This raises a slew of security and networking concerns, as well as the possibility of additional costs. Providing access with a PEP and a policy is significantly easier, quicker, and more secure.

Target-initiated access

The notion of a target-initiated action is introduced in our following example. So far, we've approached our conversation and policy model from the standpoint of an authenticated subject using a PEP to access a resource. That example, if utilizing a connection-oriented protocol like TCP, the subject's device will start the connection or network traffic (if using a connectionless protocol such as UDP). This pattern may be seen in the preceding cases, which demonstrate a user contacting a web server and a web server accessing a database. However, certain apps and networks use reverse communications, which implies our Zero Trust system must support it as well in order to achieve our aim of safeguarding all communications through policies. We still have an authenticated subject and access regulated by a PEP, but the network traffic is launched by the policy's target, and the traffic or connection is sent to the subject. Let's look at an example to put this into context:

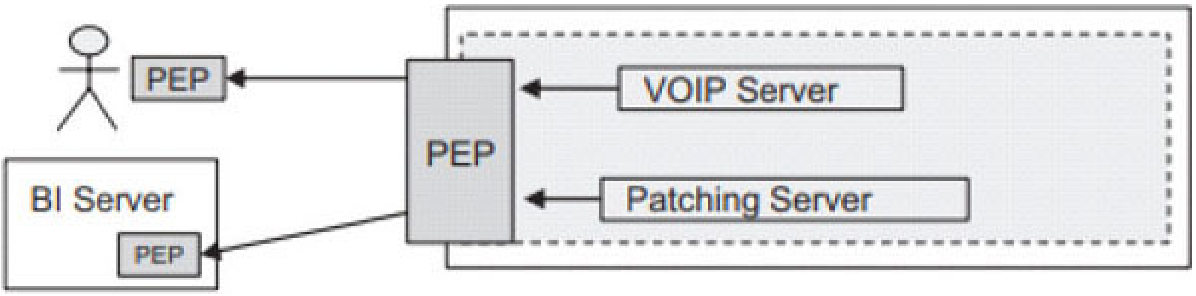


Figure 17.4: *Policy Scenario—Target-Initiated, Enclave-Based*

The Zero Trust system is deployed utilizing an enclave-based deployment strategy, with the PEP safeguarding its on-premises data center network (see *figure 17.4*). For voice communications, the organization employs softphones on user devices, and the protocol mandates that calls be started from the VOIP server to the user's device (which is running a local user agent PEP). In addition, the company has a **Business Intelligence** (**BI**) analytics server running in a remote location, which is an authenticated subject with a local PEP. Their infrastructure operations need that their internal patching server connect to the distant BI server on a regular basis in order to apply OS upgrades. The network traffic must be routed via (and managed by) the PEP in both scenarios represented in *figure 17.4*. From a technological standpoint, this places certain constraints on the Zero Trust platform and policy architecture. This scenario can be easily supported by some Zero Trust deployment models, such as enclave-based and resource-based models, which typically use a direct connection between user devices and PEPs. Solutions based on a cloud-routed deployment strategy usually have trouble with this. In our following example, we'll look at the microsegmentation deployment methodology.

Microsegmentation

Remember that the resources being accessed in the microsegmentation deployment architecture are authenticated identities, just like subjects. Therefore, the access and policy models will become more balanced. The figure 17.5 shows how the web server and the user both begin connections to the database server, and all three are authorized entities (the System Backup server will be discussed later). The consequences are that, while subject criteria and objectives would remain, the policy model in such a system will need to be slightly different. Because the web server and database server are both identities in figure 17.5, they are both authorized and have comparable sets of characteristics:

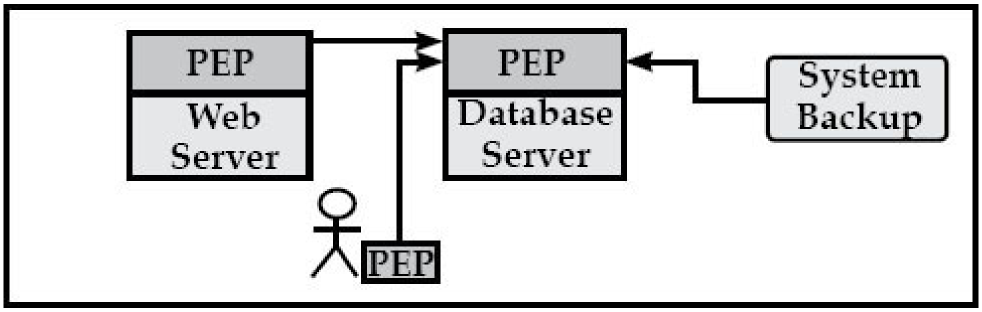


Figure 17.5: *Policy Scenario—Microsegmentation*

From a technological standpoint, this places certain constraints on the Zero Trust platform and policy architecture. This scenario can be easily supported by some Zero Trust deployment models, such as enclave-based and resource-based models, which typically use a direct connection between user devices and PEPs. Solutions based on a cloud-routed deployment strategy usually have trouble with this. In our following example, we'll look at the microsegmentation deployment methodology.

Flows of policy evaluation and enforcement

The logical system flow of policies across a Zero Trust system is depicted in figure 17.6. The PDP evaluates the set of policies in the policy store using the set of characteristics for the identity, device, and system as input. The evaluation's findings are policies that have been awarded to this topic for the duration of this session. The information on the individual to whom this has been granted, as well as information about the action, target, and condition, is included in these findings, which are sent to the PEPs. The PEP may need to evaluate the provided policies further by examining the metadata linked with prospective targets to see whether one's match:

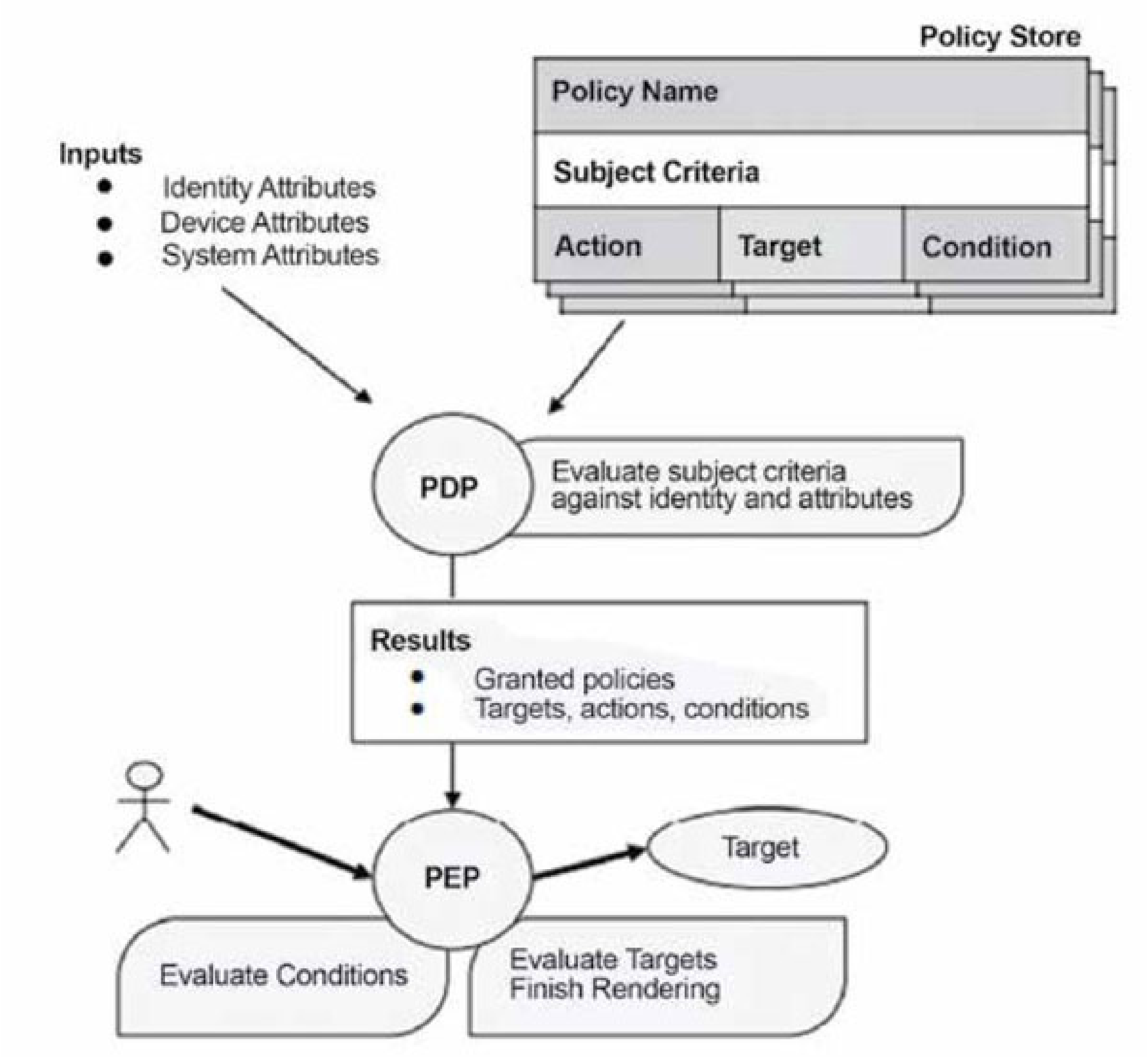


Figure 17.6: *Policy Evaluation and Enforcement*

In addition, the PEP is in charge of implementing any access-time limitations in the policies that have been authorized. The actions that the Zero Trust system performs on policies as they pass through the PDP and PEP are depicted in *figure 17.6*. While this demonstrates what the system does, we also need to explain when it does it. We touched on this in our prior chapter on security orchestration, when we discussed the basic triggers that cause actions to be taken in a Zero Trust system. The *figure 17.7* depicts these triggers and what they do when they are activated within the PDP and PEP:

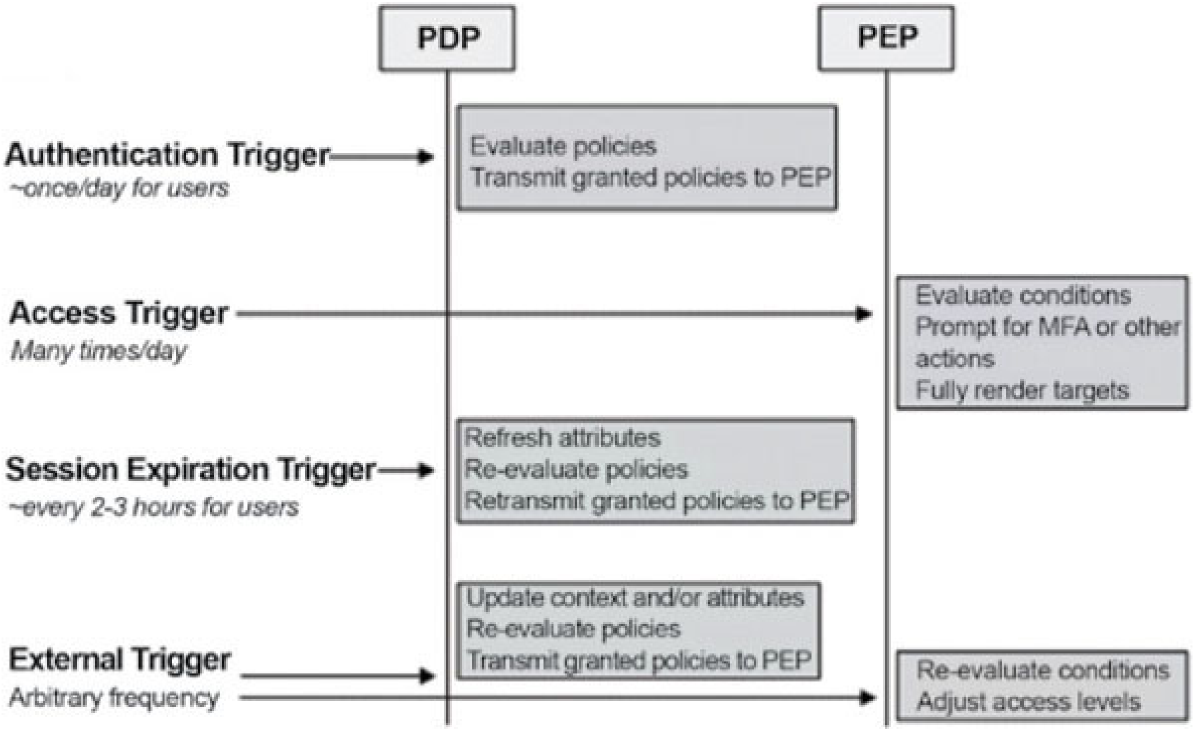


Figure 17.7: *PDP and PEP Triggers*

Authentication trigger

To be permitted any access, identities must first authenticate with the PDP. The authentication trigger will result in the policy assessment flow as illustrated in *figure 17.6*, and most Zero Trust solutions will be connected with an organization’s identity provider (rather than operating as their own IdP). Your company will be able to choose how often identities are validated into your Zero Trust system, as well as how transparent this is to end users. This selection will be influenced by a number of factors, including your desired use case and authentication techniques. We’ll go over this in more detail in Chapter 18*: Zero Trust Scenarios*, but if your system protects all of their access and is essential for them to be productive, you may want users’ devices to authenticate instantly when they login into their desktops at the start of their workday. Some businesses, on the other hand, may opt to begin their Zero Trust journey with a VPN replacement use case, in which users actively login into their systems only when they need to access certain distant services. The authentication lifecycle for system identities (non-person entities) will, of course, be different. Because these forms of identities are frequently used indefinitely, there may be no logical flow leading to regular authentication. The session expiry trigger mentioned in the next section will be more important in these instances.

Trigger for gaining access

When identities access a target, the access trigger of each policy is triggered. Depending on the deployment architecture, the capabilities of the PEP, and the kind of network protocol, different Zero Trust implementations will tackle this in somewhat different ways (for example, connection-oriented or connectionless).

Conditions may be evaluated for every network packet, every new connection to a target (if appropriate), or on a regular basis in some systems (for example, every 5 minutes). Regardless of frequency, the PEP must be capable of evaluating and enforcing variables such as time of day, external elements such as the status of a service desk ticket and asking users for any necessary interactions such as step-up authentication. The PEP is also in charge of completely displaying any targets, which means it must scour its surroundings for resources that meet the relevant metadata criteria, such as having a specified label value.

Trigger for session expiration

We haven't explicitly defined a session in this book, which was done on purpose. Sessions might mean many different things in different Zero Trust deployment models and platforms, making it difficult to utilize this phrase correctly. What important is that your Zero Trust system provides a logical idea of a session, which is a period of time after an identity has authenticated and may actively access protected resources during that time. Sessions must have a finite duration, and at the end of that lifespan, the system must refresh, obtaining new characteristics, re-evaluating rules, and communicating any changes in access to PEPs. Users may or may not see the update; this should be adjustable in your platform's policy model. Keep in mind that different sorts of qualities change at different rates, thus some will match better with being refreshed when the session is repeated. Session lengths should be considered and established in light of aspects such as risk level, use case, and identification population. Depending on how dynamic the environment and user population are, a session time of roughly 2–3 hours appear appropriate for users. That's also the maximum frequency at which consumers will accept being asked for MFA, however, once per day may be more acceptable in some situations. Season lengths for non-person entities are significantly dependent on the use case and how much your services and surroundings change. A session time of 24 hours may be adequate in some instances, but in more dynamic system contexts, anything in the 2–3-hour range would be preferable.

Keep in mind that a lot will depend on your Zero Trust platform's capabilities as well as the overhead involved with a session refresh. Remember that the most dynamic qualities should be assessed as conditions, which the PEPs should be able to refresh numerous times (even practically constantly) throughout an active session.

Trigger from the outside

One of the most important components in the success of a Zero Trust programme, in our view, is the degree to which the underlying technology platform allows and facilitates integrations. This was covered in the security orchestration chapter, but it's worth mentioning again here. Zero Trust solutions, in particular, must provide an API via which external systems can trigger a refresh.

The scope of the refresh will vary depending on the implementation, but it's critical that it includes the properties that are relevant to the external system that's starting it. Remember how we looked at an example of this in detail in *Chapter 11: Security Operations*?

Conclusion

We started with the logical components of Zero Trust policies—subject criteria, actions, goals, and conditions—in this chapter. We also looked at characteristics and their significance in the policy. Then we looked at things from a deployment and flow standpoint, looking at a variety of policy situations as well as the policy assessment and trigger lifecycle. It should be evident that the image we're painting is of a really dynamic and responsive system that relies on IT and security components working together in real-time. For businesses, this may need a cultural or technological shift, and it's critical to recognize this as part of your Zero Trust journey. It's also essential to remember that, while the principles and guidelines we've covered in this chapter are relevant to a wide range of Zero Trust platforms and architectures, there will be a wide range of capabilities in practice.

There are several distinct types of PEPs, each with differing enforcement capabilities across networks, apps, and user agents. As a result, when selecting a Zero Trust platform, ensure that you have a thorough understanding of its architecture and capabilities so that you can design your policies, as well as their lifecycles and flows, to best align with the capabilities (and strengths and weaknesses) of your chosen platform.

In the end, you want a Zero Trust platform that can seamlessly combine internal and external characteristics, as well as internal and external mechanisms for obtaining updated contextual information and making access choices based on relevant and descriptive regulations.

Chapter 18

Zero Trust Scenarios

We've looked at a lot of various facets of business security and IT infrastructure in this book. We've looked at things from a technical and architectural standpoint, and we've discussed a variety of use cases along the way. We'll look at seven potential scenarios in this chapter and talk about how to evaluate and approach them for inclusion in your Zero Trust programme. This isn't a full list of use cases, but it includes the majority of the most common circumstances.

Our objectives for this chapter are to offer you with an awareness of how and when these various scenarios could apply in your workplace, as well as pertinent advice on how to handle them.

Of course, these eventualities must also be considered from a deployment and operating standpoint, which will be covered in *Chapter 19: Creating a Successful Zero Trust Environment*. Finally, we're not going to spend much time here defending these situations for the purpose of brevity—hopefully, if you've made it this far, we've already persuaded you of that. Let's get started by replacing a VPN, which is one of the most prominent Zero Trust use cases. In *Chapter 9: Virtual Private Networks*, we discussed VPNs, their flaws, and the comparative advantages that Zero Trust delivers. We'll briefly recap this use case in this part to set the stage for a discussion on how to approach a Zero Trust project centered on an enterprise VPN (remote user access) use case. It's worth noting that we're looking at two possibilities that are related:

- Using a Zero Trust solution to replace a current VPN.
- Zero Trust deployment for a new remote access scenario

While the technical factors in these two scenarios are similar, they should be treated from distinct viewpoints in terms of reasoning and decision-making. Because there will be fewer limitations and obligations in place, new initiatives frequently represent simpler and easier options. In contrast, in a VPN replacement situation, a reason for replacing an in-place and operating VPN system is required.

This isn't to imply that this is a major roadblock—we've seen a lot of VPN replacement projects—but it does mean that security officials must be prepared to explain and justify the choice and project from a variety of angles, including security, technical, operational, and financial. It's worth noting that we highly advise enterprises to replace their VPNs with a Zero Trust solution for a variety of reasons. Let's take a quick look at the architectural distinctions between regular VPNs and a Zero Trust architecture, which were discussed in *Chapter 9: Virtual Private Networks* and are summarized in *figure 18.1*. It's important to note that this scenario is just concerned with giving secure access to services to distant users:

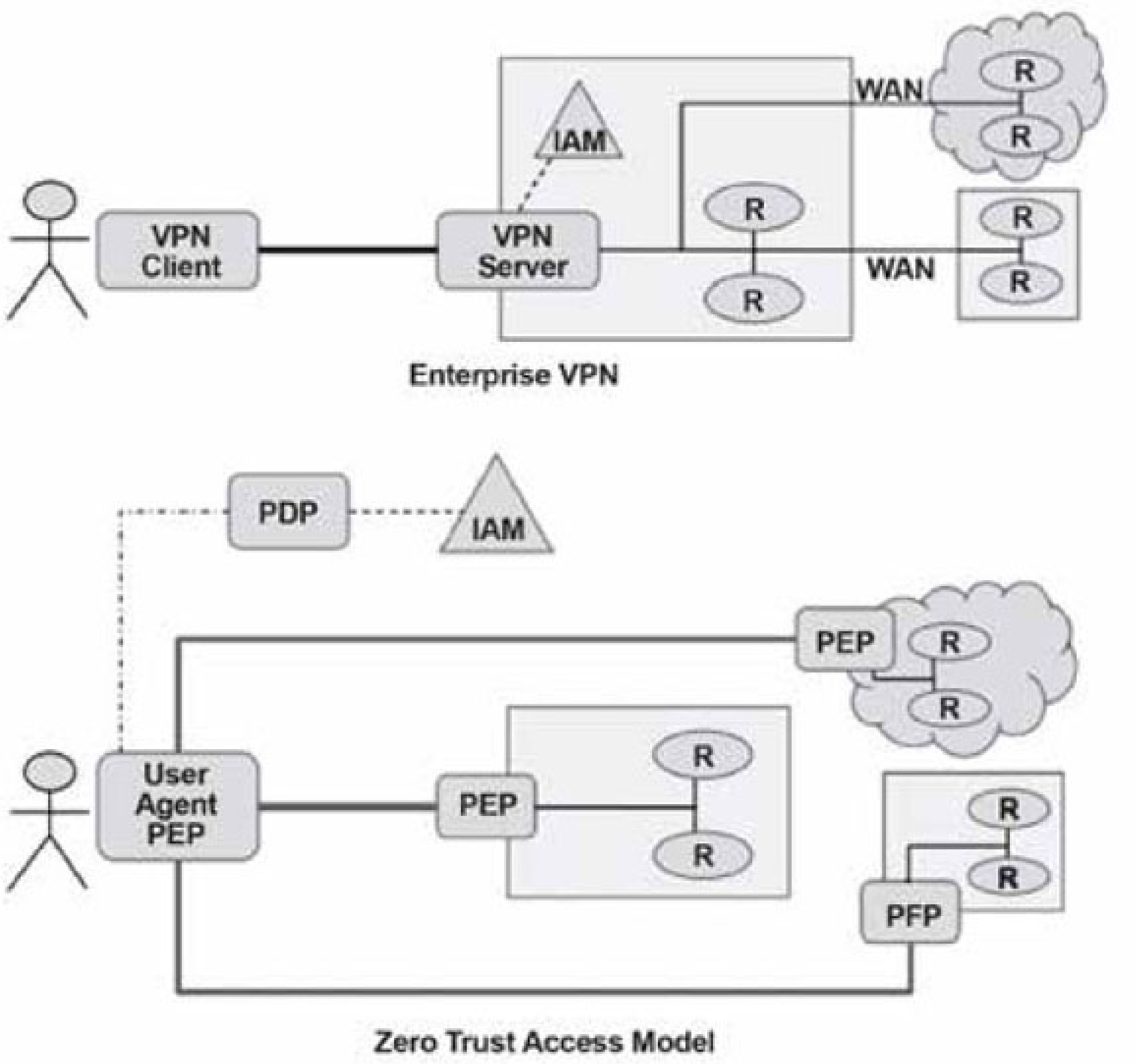


Figure 18.1: *Enterprise VPN and Zero Trust Architectures*

Traditional VPNs can only create a single secure network tunnel from the user’s device to the VPN server, which then ends the secure tunnel and allows network traffic to enter the private network region. VPNs maintain a perimeter-based network paradigm, requiring all dispersed resources to be linked to the enterprise’s core network via a wide-area network (WAN). Alternatively, when users need to access resources in multiple regions, they will have to manually switch VPN connections. Zero Trust systems, on the other hand, will create many secure connections to dispersed PEPs so that users can access them invisibly. (This applies to both cloud-routed and enclave-based approaches.) It may or may not be true for microsegmentation or resource-based models, depending on the implementation details.)

Considerations

We'll look at a few different aspects in this section to help you find potential Zero Trust VPN projects.

Resources

Examine the quantity, nature, location, and worth of the resources in question. How important are they to the business? If this is a replacement, how are they now accessed, and what are the present VPN's headaches or pain points? In general, Zero Trust solutions outperform VPNs, especially when dealing with remote resources. They can also be used to safeguard resources in places or situations where the company cannot set up a VPN entry point, such as a third-party network.If you have a set of resources that is extremely dispersed or dynamic, you should consider using a Zero Trust approach—recall the dynamic target rendering from our policy model chapter.

User experience and users

Who are the users who are using the VPN right now, or who require access to these additional resources? Are all of the users online? Was this remote user access solution rolled out quickly (perhaps with some known flaws or compromises) in reaction to the COVID-19 work-from-home shift, for example? Are on-premises users gaining access to these resources via a different security paradigm, such as firewall ACLs?

There are frequently solid reasons to employ Zero Trust in certain situations, such as to address security or operational concerns generated by a quickly established VPN. If resources were recently deployed, it's possible that only distant VPN users have a safe access channel, and your company need a solution for on-premises users. Finally, Zero Trust solutions, which are meant to protect access to all resources for all users, can replace siloed solutions like different rules and access methods for remote and on-premises users.

In next chapter, we'll go over this in more detail.

Providers of identity

Some VPN deployments do not interact with business identity suppliers; in these circumstances, a Zero Trust deployment can bring significant value immediately. Security teams may eliminate an identity silo within their VPN by linking remote access user authentication to their business identity provider. This avoids any effort required to maintain that silo in sync with their parent provider, such as responding to Join, Move, and Leave identity lifecycle events. Even if a VPN relies on a corporate IdP, a Zero Trust solution will enhance it by implementing fine-grained and context-sensitive access controls.

Many Zero Trust solutions also enable numerous identity providers of various sorts, allowing distinct user groups to authenticate against alternative IdPs or current authentication protocols to safeguard legacy systems.

Networking

You must have a thorough awareness of your company's network structure, data flows, and where the protected resources are kept. With this knowledge, you'll be able to make well-informed judgments and suggestions when it comes to switching from VPN to Zero Trust access. Begin by determining the location of VPN concentrators (entry points), the networks to which they give access, and how dispersed resources are accessible from a network standpoint.

Determine if users are only accessing resources through a single entry point into a corporate network, as we discussed in the introduction to this chapter. Even in this basic scenario, Zero Trust may bring benefits such as enhanced performance and stability, improved interaction with identity providers and MFA, and, of course, fine-grained access restrictions. VPNs are frequently used by teams or projects that require access to distributed resources, and here is where Zero Trust thrives (as long as your chosen implementation and deployment model supports multiple concurrent connections to distributed PEPs). Consider this an opportunity to ask the networking or application teams “what if” questions. “What if consumers could use both of these resources at the same time?” “What if we could connect access to a business process like a service desk ticket?” “What if we could run more thorough device posture checks before granting access to users?” These are great questions to start a dialogue with those teams and enlist their support for your Zero Trust initiative. There are a few more questions you should ask your networking team to assist you plan and advocate for your Zero Trust deployment. Find out what sorts of remote access policies (ACLs) your existing VPN uses, for example. How big or little are they? Your Zero Trust project can give increased security and lower risk by drastically decreasing network access without losing user productivity if they grant very broad network access, which is relatively frequent. Determine if your project can handle any lingering compliance concerns or audit results. If your VPN restricts network access, see how well it performs in terms of operational efficiency and user productivity. In all but the most static contexts, this is likely to generate operational effort as well as user irritation. Your Zero Trust solution should be able to enforce similar (if not more stringent) access control limitations via automated policies, saving your IT and operations teams time and effort. Finally, find out how your company uses any wide area networks. Organizations generally incur significant expenditures as a result of this, and Zero Trust solutions can minimize (and, in some cases, eliminate) WAN usage.

Recommendations

A VPN replacement or alternative is a frequent first Zero Trust project, and they’re usually a smart place to start. The advantages are obvious, and a Zero Trust solution can typically replace the functionality of a regular VPN. We do advocate a phased rollout, keeping in mind those user groups who may need to keep both Zero Trust and VPN access for a while. On an end-user device, these solutions can typically coexist in peace, but they can’t operate at the same time since they’ll fight at the networking level. This might be a problem if you have an “*always on*” VPN, or if you intend to use Zero Trust in a similar fashion. One last suggestion for the VPN replacement situation is to take a close look at the tools and procedures that have been created around the VPN tool's scope and capability. Some businesses, particularly those with older VPNs and equipment, may have created a "*web*" of interconnected technologies. This might be a significant stumbling block for a gradual Zero Trust implementation. For example, one organization we dealt with had a classic VPN, which logged certain events into the user's Windows event log. They'd created a collection of "*glue*" programmes that monitored the Windows event log and performed network setup operations in response to those occurrences. Because that component was maintained and controlled by another team inside the company, modifying these tools was an extra work that caused a delay in the project. So, be aware of how your corporate IT environment works, and ask plenty of questions all the way up and down the IT stack, as well as throughout your whole IT and business process ecosystem. You might be surprised by the extent to which the company has become reliant on specific tools or workflows. Some of these may be roadblocks to Zero Trust adoption, while others may be present issues that your project can address.

VPNs generally come with a slew of issues, which is why they're such a good place to start when it comes to Zero Trust projects.

Access by a third-party

Third-party access is another ideal candidate for Zero Trust since it's a common source of difficulties and risk for businesses, and there's a clear difference and advantage to using Zero Trust vs. regular third-party remote access. Let's start with a definition: a third party is a non-employee individual with whom the company has a legal connection and who requires valid access to the company's network and private resources for the purposes of this discussion. Specifically

- The persons can be recognized;
- The resources they require are well-known and identifiable;
- They need access to private firm resources (if all they need is internet access, they could just use the guest network when on premises).

Note that this scenario does not include full-time contract (non-employee) workers; in our experience, these people are treated much more like regular, full-time employees in terms of IT. A contract programmer on a six-month assignment, for example, may not be a corporate employee, but he or she will normally be given a company-managed device and will be enrolled in the enterprise's identity management system. They should be controlled in the same way as workers in terms of security, but with far more limited network access.

Consider a few instances of the types of third parties that may be involved in this scenario. These are frequently linked to outside businesses having particular expertise in a given field that would be impractical to have in-house. A company that monitors, maintains, and services building HVAC systems, for example, is a classic third-party access risk. HVAC vendors demand periodic access to these systems, which are often on the business network, in order to maintain them working efficiently. Another example is a company that needs access to an on-premises financial management system for outside financial auditors. These are the categories of third-party users who require additional security restrictions. *An organization cannot impose internal policies on external actors (for example, clients or general internet users), but it may be allowed to enforce some Zero Trust-based policies on non-enterprise users having a specific connection with the organization*, according to the NIST Zero Trust paper.

These users must be validated, and their network access must be limited to the bare minimum, according to our Zero Trust standards. Organizations have traditionally utilized VPNs to offer remote access for third parties, and VPNs, of course, have all of its flaws when it comes to third-party access. Furthermore, because these third-party users are not workers, they are not using enterprise-managed devices by definition.

This implies that the company cannot require or depend on the device's security posture, making it even more critical to enforce security controls around the device's network access. One last barrier is that security teams cannot mandate the installation in general.

of any particular programme installed on such devices. With the rising prevalence of **Bring Your Own Device** (**BYOD**) and the acceptance of using personal mobile phones or tablets for work-related tasks, this is a little less absolute than it used to be. For example, even though remote access is a requirement of their employment, a third-party user may be unable to install remote access software on an enterprise-managed laptop.

However, installing remote access software on a personal tablet or a BYOD device and using it for business duties is becoming increasingly common.

Even if the third-party user can install remote access software on their device, it's doubtful that they'll agree to the installation of more invasive endpoint management or security software, thus include these devices in your enterprise's security or IT management system is unrealistic. Accept that these systems and devices may not match your security criteria, and utilize Zero Trust to enforce the concept of least privilege, as well as MFA. In the "Recommendations" section, we'll go through this in further detail.

Considerations

Third-party access is an excellent candidate for a Zero Trust project in general, and it may sometimes be a suitable starting point. These users are usually highly well-defined, and their access to resources is usually confined to a small and unchanging set. They're also a concern because these users are accessing enterprise-controlled resources from devices that aren't managed by the company.

Architecture

Third-party access network design will most likely resemble that of your VPN; in fact, it's very possible that these individuals will utilize your current workplace VPN. What's critical to understand is how and where these users are coming into the network, as well as how their network traffic traverses the company to reach their target resources, much as in the VPN use case. The kind and location of these resources should influence where your PEPs are placed, allowing you to avoid having a large amount of third-party user traffic transit your network. As always, the concept of least privilege applies, and your PEPs should prohibit these users from gaining any unwanted network access.

User experience and users

When compared to workers, the user experience of third-party users may be less significant. This is especially true if access is only required on an as-needed basis rather than on a daily or all-day basis. Employees, for example, may want visible (always-on) access to Zero Trust–protected resources, whereas third-party users may not. Having said that, it's evident that you shouldn't make it impossible for them to get in. Agent-based and agentless access are frequently supported by Zero Trust systems, and third-party access is a common use case when agentless access is required. Agentless access may be a possible alternative depending on the type of resources being accessed and the network protocols being utilized. Web-based apps are often easy to reach using an agentless paradigm, however non-web (non-HTTP) applications might be difficult. There are other solutions, albeit at a higher expense, if a Zero Trust agent is theoretically required on user devices but the third party refuses to install it.

For example, the company may host a virtual desktop for third-party users, where the Zero Trust agent would be installed. Alternatively, the company may set up a controlled device that other parties could use to gain access to the Zero Trust-protected environment.

Recommendations

From the standpoint of user authentication and identity management, we propose that your Zero Trust system use the third party's corporate identity management system for authentication if at all feasible, but only if you are confident in their maturity and identity lifecycle procedures. If not, instruct them to use an IdP that you control, such as your main corporate IdP or a smaller, simpler one devoted to third parties. Any Zero Trust solution should be able to authenticate multiple user groups against various IdPs. We also propose that you implement **multi-factor authentication** (**MFA**) for these users every time they try to access your services. This type of step-up authentication should be deployed using your Zero Trust system and an MFA provider that you control.

This guarantees that you can enforce your security standards regarding authentication frequency and type, as well as eliminate the possibility of third-party users exchanging credentials (which is a common occurrence). Contextual access constraints, such as geolocation, should be enforced by your Zero Trust system, as should fine-grained access regulations that limit user access to the bare minimum.

Third-party access is often only provided for a definite and well-defined set of objectives; thus these policies should be simple to design. When practical, we also recommend that you tie your third-party access policies to a business process to further restrict (and document) this access. Many Zero Trust systems, for example, allow you to create policies that limit access based on the existence and status of a service desk ticket. This strategy will work effectively in circumstances where third parties only require access on a regular basis, since it ensures that all access is sought, authorized and provided for a limited duration. Finally, even on-premises third-party users must access resources from within the Zero Trust model if an organization has already made the move to Zero Trust and has a "*café-style*" network. That is, any third-party users who are physically present in a corporate facility will receive the same limited access they do while working remotely. This is a key feature of Zero Trust: third-party in-person network access no longer puts the entire company network at risk.

Cloud migration

Migrating apps and services to cloud platforms is an important aspect of today's company IT and application development, and it may take many forms. Because of the strength of these platforms and the accessibility and stability of network connectivity, this is an inexorable trend, which is why Zero Trust projects and leaders must embrace it and educate their business and application development colleagues about it. In an ideal world, security teams will have a Zero Trust platform in place, as well as an organized menu of ways and permitted components, allowing app owners to easily embrace the cloud.

Migration categories

Of course, "*cloud migration*" is a broad term that encompasses a variety of activities that are dependent on a variety of conditions.

Forklift migration

The application is "*as is*" migrated from an on-premises physical or virtual environment to an IaaS environment in this scenario. That is, no modifications to the application logic, topology, or technology have been made. As a result, the identical programme is now operating in a new location. Because the application's structure and interdependencies are preserved, this migration is faster and easier, but the advantages are limited. This migration should not necessitate any application development modifications; it should only necessitate reconfiguration, and it is ideally suited to COTS software that the organization has licensed but cannot modify.

Refactor the application

The application is transferred to an IaaS environment in this case, but certain technical or structural changes are made to take advantage of the new cloud platform. The programme might be changed to utilize a cloud-native database or a cloud-based identity provider, for example. Alternatively, portions of the application's deployment or operations infrastructure (such as the web server or a logging server) might be rehosted on a cloud-based alternative. This migration necessitates technical or development modifications to the application, although it usually yields reasonable results. Some COTS software will help with the migration in subtle ways, for as by allowing users to access a cloud-based database.

Rewrite the application

This strategy is the most technically challenging, but it has the most potential for utility. Application developers may use this paradigm to fully rethink their application design, including adopting a "*radical*" approach to incorporate current components like containers, PaaS, microservices, and NoSQL databases, among others. Developers may be able to reuse components of the application logic and data model depending on the present application architecture to help speed things up. This method is ineffective for COTS applications.

Adopt SaaS

Organizations are migrating from on-premises systems (custom or COTS) to cloud-based SaaS apps using this strategy. Of course, this necessitates a complete reorganization of the application's structure and access restrictions. Some on-premises application logic may be reusable, especially if the organization is adopting the SaaS version of its on-premises application. In order to accelerate the value of their SaaS service, enterprises should be able to import part of their application data.

Many (if not all) cloud migration projects are ideal candidates for Zero Trust because they include changes to security, network, and architecture, and so provide a chance to adopt a contemporary, cloud-friendly security platform. Zero Trust systems, in particular, may benefit from the broad range of APIs provided by cloud platforms due to their dynamic and context-sensitive nature.

Considerations

Each of these four migration scenarios represents a unique opportunity to implement Zero Trust, which may significantly increase the value and security of these in-motion apps. Let's take a look at them from an architectural standpoint.

Architecture

Consider our comments in the chapters on IaaS, PaaS, and SaaS, where we discussed the network access restrictions and architectures associated with those models. Examine your organization's planned or ongoing cloud migration architecture and methodology and make changes to verify that it is compatible with your Zero Trust network topology and access restrictions. And, based on your selected cloud migration strategy, ask yourself and your company the following questions.

Forklift

Is the app self-contained, and are all of its components being forklifted to the cloud? Because most apps aren't completely self-contained, how will data flows in and out be managed? What role may your Zero Trust PEPs play in this? Will the application's (non-user) components be contained within an implicit trust zone? Is that risk acceptable under your new security approach, if so? If not, how will they be verified and granted access through a PEP?

Application refactoring

What is the present and anticipated network topology, in addition to the Forklift questions? What is different about the component interactions these days? What are your options for influencing modifications to the application's design?

Rewrite the programme

In creating a new application architecture, how much will the application team be "beginning from scratch"? What will happen to current application components (both functional and data)? Is it possible to integrate the new architecture with your Zero Trust platform? Will it be necessary to cohabit the old and new versions for a period of time? Will they have to share data if this is the case? How will this be accomplished?Finally, can the application be designed in such a way that it consumes Zero Trust policies from the PDP and becomes a PEP application?

Adopt the SaaS model

Because the new platform is not under your control, this is clearly a different strategy than the prior three. Because the destination is known, this may be an easier transfer from a security and network standpoint. Examine the SaaS platform from a security standpoint and assess whether it makes sense to deploy Zero Trust security to this SaaS environment using the standards we provided in our previous chapter.

User experience and users

As a result of their relocation to the cloud, these newly migrated apps will most likely have various network access models. The end-user experience may be disrupted or challenged as a result of this. Your Zero Trust solution can frequently reduce friction by providing users with transparent and secure access to cloud-based applications while also implementing dynamic and context-sensitive access controls.

Recommendations

We strongly advise that you cooperate with the application owners and integrate Zero Trust as part of the migration and deployment plan when these apps migrate to the cloud. The sole possible exception is the use of SaaS apps, which may not need Zero Trust in all environments.Finally, be proactive and collaborate with your coworkers who are application owners.

Introducing them to your Zero Trust platform design and strategy can really help speed up cloud migration efforts.

Access from one service to another

Service-to-service access control is unquestionably a valid, beneficial, and critical Zero Trust application. For good reason, many corporate Zero Trust solutions begin with and focus on user-to-service access. Users and servers have extremely distinct risk profiles and live in quite different worlds.

Users

- Are untrustworthy and unpredictably volatile
- Use insecure, unmanaged networks to run their devices
- Are mobile, allowing access from a variety of locations.
- They have a proclivity for misplacing their equipment.
- Frequently repeat passwords or use weak passwords
- You can’t function with a whitelist of Internet destinations without affecting user productivity.
- Receive emails containing phishing links and click them on occasion
- Install unmanaged and arbitrary software on devices

Users, in other words, are unpredictable, innovative, and prone to making mistakes. Servers (and the services that operate on them) on the other hand, are (or should be) the polar opposite:

- They can only be used on enterprise-managed networks.
- Are more trustworthy—IT should know, manage, and control 100 percent of the services operating on every particular server.
- Don't go to random Internet addresses—the collection of internal and external network addresses can theoretically be known and whitelisted.
- Do not open emails containing phishing links.
- They don't get lost in pubs or restaurants.

Indeed, servers are trusted enough that many Zero Trust systems contain a network segment, hidden behind a PEP, where servers connect outside of the Zero Trust environment's control—the implicit trust zone, which we've explored throughout the book.

To be clear, we're not trying to discourage you from using Zero Trust for a service-to-service use case; we're only pointing out that user-to-service poses a larger risk. Nonetheless, service-to-service access restrictions should be included in every Zero Trust endeavor, and they may even be one of the first use cases. Let's take a look at the value and advantages that Zero Trust may provide in this situation.

Most significantly, Zero Trust adheres to the concept of least privilege, which is critical for limiting the attack surface and explosion radius of any successful assault. There is a reduction in risk as a result of this. It also ensures *top-down* visibility and control of service-to-service interactions, as all communications are expressly permitted by rules. That is, security and networking teams are no longer reliant on identifying interactions between services using a certain protocol. Instead, because it functions on a default-deny basis, the Zero Trust system assures that every service-to-service communication occurs only if and only if it is expressly approved by a policy. Because all service-to-service interactions must be enabled by an approved policy, this has an intriguing effect—it actually acts as a sort of referential integrity for the network—it assures that this communication is expected by deployment systems and procedures. Because unanticipated communication paths will be closed, the development and deployment process will be more mature and predictable. While this may appear to add to the friction, it will be more than compensated in terms of better dependability, automation, security, and robustness. It also assures that deployed services are documented and catalogued, removing the problem of *don't touch that server, we have no idea what it does*. While this may appear to be sufficient to justify the service-to-service use case, it also has other advantages. Zero Trust results in a decrease in total risk and, as a result, an increase in compliance. Many compliance-driven measures, particularly for high-value workloads, need improved network segmentation. In the event that apps use unencrypted protocols, Zero Trust ensures that network communication is encrypted. Finally, because Zero Trust systems can dynamically and automatically adapt to changes inside the set of protected resources, organizations may use high-velocity development methods (such as DevOps, which we'll talk about momentarily) without losing security.

Considerations

Microsegmentation appears to be the logical choice when it comes to Zero Trust models in the context of service-to-service, and it may well be the greatest match for settings where all servers have IDs and can be verified. Because, in the microsegmentation model, all servers are identities (Zero Trust subjects), and access control mechanisms tend to reflect this service-to-service symmetry, this is a requirement. The enclave-based and cloud-routed models will also work for this use case, and in fact, they may be a better fit for contexts where Zero Trust is new. These approaches provide additional flexibility, particularly in situations where certain identifiable and authenticated services (subjects) need to access distant services that are targets protected by a PEP but are not Zero Trust subjects. In reality, asymmetric service-to-service, where one service is an authenticated identity and the other is not, but resides behind a PEP, is likely to be a typical server-to-server scenario in many deployments, as shown in *figure 18.2*:

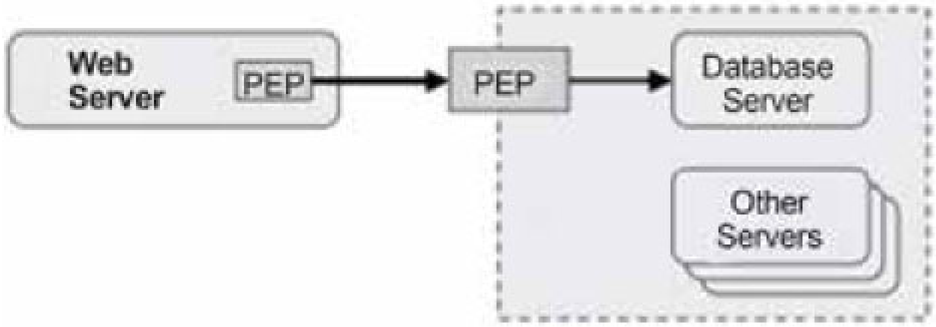


Figure 18.2: *Asymmetric Service-to-Service*

This paradigm is a useful alternative to “pure” microsegmentation, which requires every service to have an identity and may not be appropriate for some organizations or architectures. This method may also be used to secure service-to-service access across many networks, which is particularly important for dispersed application components that may emerge as a result of a cloud migration. Because there is an inherent requirement for a security overlay that normalises the access control model in use, cross-network service-to-service access control is a natural use case for Zero Trust. Actually, there is one more service-to-service option worth mentioning, which is the use of a non-identity access control method akin to that used in the Internet of Things. In this paradigm, neither of the services are verified identities, as we discussed in Chapter 16.That is, you may treat your connection-initiating services like an IoT device, with access constraints based on weaker kinds of identity and authentication like MAC address, IP address, VLAN, or switch port. This is possible, but it comes with some drawbacks, as we saw in Chapter 16. As a result, we don’t advocate using this strategy for service-to-service—far it’s preferable to verify at least one of the identities.

Recommendations

Determine where you have servers communicating across network or domain boundaries to identify good candidates for the service-to-service use case. This will be a natural site to put a PEP, given the traffic is passing a network border. As a result, solving this problem can be rather straightforward. Depending on the network configuration and how difficult or easy it is to isolate the servers behind a PEP, targeting peer servers on a single internal LAN may be more difficult. High-value or compliance-driven server isolation, on the other hand, may be a valid reason to prioritise this situation, particularly if there is a significant requirement from a risk or auditing standpoint.

These factors can be a catalyst for implementing the necessary network \sand access modifications. Explore your environment as you consider this use case and attempt to find services that might be a good fit—in particular, high-value, well-understood and well-controlled services that are maybe extremely dynamic and challenging to protect with present solutions. Automated Zero Trust policies may be a tremendous help here, adjusting access to reflect changes in your server environment, without needing manual work. You should also keep in mind that many servers run numerous services, and you can opt to put only part of them behind a PEP while leaving the others alone. For example, you may use a PEP to restrict server-to-server access to a Database service operating on a specific host while still allowing non-Zero Trust users to access a web server on the same host.

Finally, examine any microservices environments that your company has implemented. A microservices environment, such as a service mesh, may not be the greatest Zero Trust candidate, as we mentioned in *Chapter 14: Infrastructure and Platform as a Service (PaaS)*, because it likely has its own internal and self-contained authorization scheme. However, if there is a clear demarcation barrier and a natural match for a PEP, service-to-microservice might be an excellent place to start.

To be effective, your policy model must allow for the definition of microservices as targets, as well as attribute and context-based access controls.

DevOps

DevOps, which is a combination of the phrases development and operations, is a novel method to application development that emphasizes cooperation between formerly compartmentalized software development and operations teams. This method, which does include cultural and procedural changes, has been shown to enable enterprises drastically boost their deployment by utilizing automated toolsets and short cycle times.

Release pace, release quality, and commercial value are all important factors to consider. Finally, DevOps is about delivering code into production rapidly and consistently. DevOps teams typically use **Continuous Integration** (**CI**) and **Continuous Delivery** (**CD**). Approaches to DevOps delivery that use a high degree of automation across the build, test, release, and deploy stages.

This automation is linked to the "*infrastructure as code*" method, in which not only the software programme, but also the virtual infrastructure on which it runs, is produced and deployed automatically, and both are specified by configuration (code) in a repository. This may appear complicated, and it is, but it has given organizations the ability to quickly bring applications to market, increase team productivity, stabilize production environments, improve customer satisfaction, and provide consistent code deployments—all of which have resulted in increased business value:

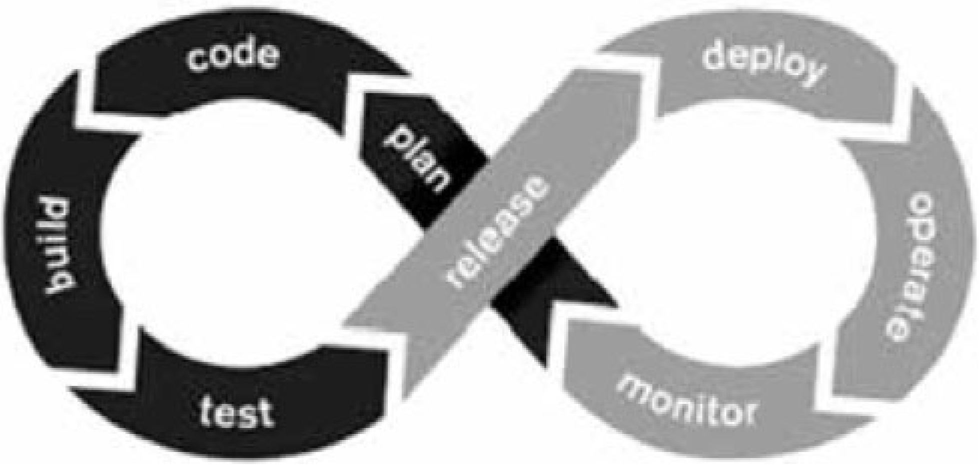


Figure 18.3: *The DevOps Cycle*

The various phases of DevOps are depicted in *figure 18.3*. The *infinity* symbol is widely (and purposefully) used to signify the continuous and never-ending nature of DevOps. Of course, where security fits into the DevOps approach is a legitimate question. The answer is *everywhere*, and it is the only right one.

In truth, DevSecOps is a phrase and a collection of methods dedicated to integrating security into DevOps. Multiple facets of security are correctly included into the software design, development, deployment, and operations with this technique. This is significant because, in the past, security was treated as an afterthought in development, which had negative consequences. Security frameworks, on the other hand, may be successfully integrated throughout the DevOps cycle when security is designed in and thought through at the forefront.

While we'll be looking at DevOps from a Zero Trust viewpoint in this section, there's a lot more to application security than Zero Trust, including static code analysis, functional security testing, fuzzing/input validation, and library vulnerability management.

Phases of DevOps

Let's take a look at the DevOps phases and see how they relate to Zero Trust.

Create a plan and a code

This is the phase when security teams should cooperate with and educate application developers about their Zero Trust architecture, capabilities, and policy model from a design standpoint. This knowledge will aid application designers in determining where they may rely on the Zero Trust platform and where they must take responsibility.

If a high-value application can rely on the Zero Trust platform, it won't need to integrate MFA, device posture checks, or geographical limitations. Additionally, application developers may be able to use the Zero Trust platform to gain extra user context, such as role or permission validation. These may be ingested and enforced from within the app, thereby turning it into a policy enforcement point.

Construct and test

As application code progresses through the build and test phases, the Zero Trust system may employ automated policies to allow access only to the relevant people and tools depending on workload characteristics. A testing workload, for example, may be automatically started and only allow access to in-progress application instances that have been correctly labelled as being in test mode.

Deployment and release

The application will be deployed in a Zero Trust environment with a full set of policy enforcement after these last steps of the release process. That is, policies govern all access to application services, which are only provided to authenticated and authorized users. Zero Trust rules may even regulate access to the production environment, depending on the degree of automation, for example, based on allowed modification windows or a legitimate Service Desk ticket.

Monitor and operate

Zero Trust will assist in ensuring the environment's stability and controlling any administrative or troubleshooting access to production apps during this time. It will also offer identity-enriched logs, guaranteeing that all access is linked to validated individuals.

Considerations

DevOps is an intriguing and important use case for Zero Trust since there are so many ways to relate it to, and receive benefit from, Zero Trust. Security and application development teams may balance and share access control methodologies and rules even with simple integration. Breaking down this conventional barrier allows Zero Trust integration to be baked in throughout the application lifecycle. The effect and value of Zero Trust in the company may be increased by designing an application component (or microservice) to consume and enforce PDP-defined policies. In effect, this allows an application to act as its own PEP in certain aspects (depending upon how much Zero Trust policy or context it can consume from the PDP).This may be weaved into DevOps cycles, where the set of policies provided to the application (and hence enforced) is changed to fit the application's current phase. Consider the use scenario we mentioned before, in which manual code release and deployment might be a security risk. Organizations may guarantee that this high-impact access is adequately regulated by adopting Zero Trust policies throughout the release and deployment phases, for example, by enforcing allowed change windows.Finally, one of the most common Zero Trust use cases is controlling access to your company's software designs and source code. These assets are obviously important, and they, like any other high-value data, require rigorous security and access control by a PEP.

Recommendations

In contrast to the conventional Software Development Lifecycle, the goal of DevOps is to provide a high-velocity, high-quality, high-reliability method of putting application code into production (SDLC). Many of today's rapidly changing settings are better suited to DevOps, where putting incremental code into production fast is frequently what creates business value. Zero Trust solutions are well suited for usage in a DevOps environment because they are intrinsically dynamic and sensitive to user, service, and infrastructure context.

Connecting a Zero Trust system to an organization's DevOps platforms allows it to automatically modify access as workloads go through the application lifecycle. Zero Trust also helps to improve and automate security in areas where human processes are still required, such as access controls based on allowed change periods.

DevOps and Zero Trust are both current and successful practises, and businesses should consider how they may work together to assist one another.

Acquisitions and mergers

Mergers and acquisitions (**M&A**) are difficult and frequently protracted undertakings that must strive to integrate two previously autonomous firms from a security and technological standpoint.

The IT and security infrastructures of these companies were designed and matured independently, adopting technologies and designs that may be incompatible (or at least difficult to reconcile). In many locations, these two firms will very probably have duplicate solutions, as well as overlapping network IP address ranges, which will almost surely cause problems—an all-too-common occurrence in our IP v4-centric environment.

Remember that, in addition to providing security, Zero Trust platforms also serve as a unifying or normalising layer on top of diverse resources and networks. This has several advantages inside a single business, as we've explored throughout this book, and it also aids in the rapid provisioning of network access during a merger and acquisition.

A Zero Trust system, in particular and tactically, can give near-immediate IT access across domains, allowing for rapid collaborative management. It can also provide accurate and secure user access to business-critical programmes such as finance management systems. Let's look at the next level of information based on this value.

Considerations

If one of the two companies already has a Zero Trust deployment in place, an M&A transaction should be a natural stimulus for expanding its use, especially if the acquiring business is the one doing the acquisition (which tends to be larger, and more able to impose its IT and security infrastructure).

However, even if the acquired firm has Zero Trust, the combined company can still use that platform to at least speed up the integration process. No other security or remote access solution can bring two distant (and frequently conflicting) organizations together as quickly, reliably, or accurately as this one.

A Zero Trust strategy might potentially save money and time by avoiding the expenditures and efforts that are generally required to integrate, normalise, or de-conflict networks. If all users and servers have the access they require through a Zero Trust system, it may not be necessary to establish a WAN to connect the company networks.

Furthermore, if the Zero Trust system includes access methods that may compensate for overlapping IP addresses on networks, organizations may not need to de-conflict them. Consider the resources users require rapid access to, where they are situated, and how they are safeguarded currently as you approach this use case. Naturally, each company will have its own identity provider, IT management, and security solutions, all of which Zero Trust can assist to standardise quickly.

Recommendations

If you already have a Zero Trust solution and are buying a company, leveraging it to speed up the transfer should be a no-brainer. If you don't have such a solution in place but the firm you're buying has, you should definitely consider adopting it to aid with the transition. At the absolute least, your staff will be able to utilize it to access the purchased company's resources. And, for example, by establishing a PEP in your company's network, you should be able to quickly expand that system to provide acquired company users access to your firm's resources. In this scenario, you should be able to utilize it to make the case for Zero Trust adoption inside your bigger organization—the acquired business has shown success with it, and you should be able to leverage it rapidly to create benefit. Last but not least, don't overlook the server-to-server scenario. Data synchronisation or export/import operations need production servers in one domain to safely interact with production servers in another domain in many circumstances. Zero-trust technologies enable this to be accomplished swiftly and securely without putting either business at risk.

Divestiture

Divestiture, in which a company spins off a portion of its business into a separate company, can be a difficult issue for IT and security, but it can also be an exciting opportunity. Part of the IT and security infrastructure will almost probably be passed down to the new firm, which may include physical assets such as hardware, networking gear, networks, and buildings. While these assets define "*brownfield*" settings, IT and security teams are usually given the authority to choose new systems and tools to fill in gaps or replace pieces that must be retired over time.

time. This should provide the opportunity (and resources) for the IT and security teams to implement a Zero Trust system in this new context.

Aside from establishing a new company’s infrastructure, there’s another component of a divestiture that lends itself to Zero Trust: the transition time. Almost every divestiture involves a business and legal transaction that must be completed before much of the technical work can begin. Even when the companies are officially split, they will remain linked by a slew of technical systems, data flows, and business procedures that can take months to untangle. Zero Trust can be used to give exact access control to important resources that were “*left behind*” during the transitional phase, allowing users and servers to continue working while prohibiting unwanted network access.

As the new organization transitions off systems one by one, access to them may be quickly cancelled by changing a policy in the Zero Trust system.

Network/network transformation with complete Zero Trust

This is an appropriate use case to end the chapter with, since it sets the stage for Chapter 19’s discussion of the Zero Trust deployment path. This scenario is a hybrid of the scenarios we’ve just discussed, yet it differs significantly from them all in certain aspects. The most significant distinction is that adopting “full Zero Trust” necessitates a change in networking mindset, namely, removing all users “off net” and requiring them to use the Zero Trust system in order to access any organizational resource. Interestingly, many businesses’ willingness to undertake this move was expedited by the rapid COVID-19-driven shift to a largely work-from-home user base in early 2020. The most significant mental adjustment is realising that the problem to be solved isn’t “remote access”—it’s simply “access.” In reality, much of the benefit of a Zero Trust environment is based on having a consistent strategy to safeguarding all access. The phrase “complete Zero Trust network” implies a broad and comprehensive scope, but in fact, you set the limitations and bounds for your initiative—not every Zero Trust journey must finish with resource microsegmentation.

In some aspects, it’s better to use the more vague word “*network transformation*” rather than the more specific term *complete Zero Trust*, which might lead to a misunderstanding. So, while you go through this process, make sure you set boundaries and have a clear picture of where you want to end up. In our experience, the following is the most prevalent way that businesses envisage their Zero Trust end state:

- All users are no longer connected to the corporate network.
- PEPs safeguard the majority of private services, with the enclave-based type being the most common.
- PEPs may be used to safeguard some SaaS services.
- Microsegmentation could be used for some services.
- There will be certain implicit trust zones where services will be available.

The improvements and advantages that we've been promoting throughout this book are, of course, the consequences of this. The removal of the trusted enterprise network increases the organization's survivability while also reducing the attack surface and blast radius. Users have "*always-on*" Zero Trust access, with dynamic and context-sensitive policies assessing whether they have enough access to be productive while adhering to the concept of least privilege. This concept guarantees that every access is provided openly by policies, improving the visibility of the organization's network and computing assets. Furthermore, the company IT and security infrastructure is linked at the data and process levels, resulting in increased efficiency and effectiveness. Take a look at *figure 18.4*, which shows the basic Zero Trust architecture model we discussed early in the book in Chapter 3: *Architectures With Zero Trust*:

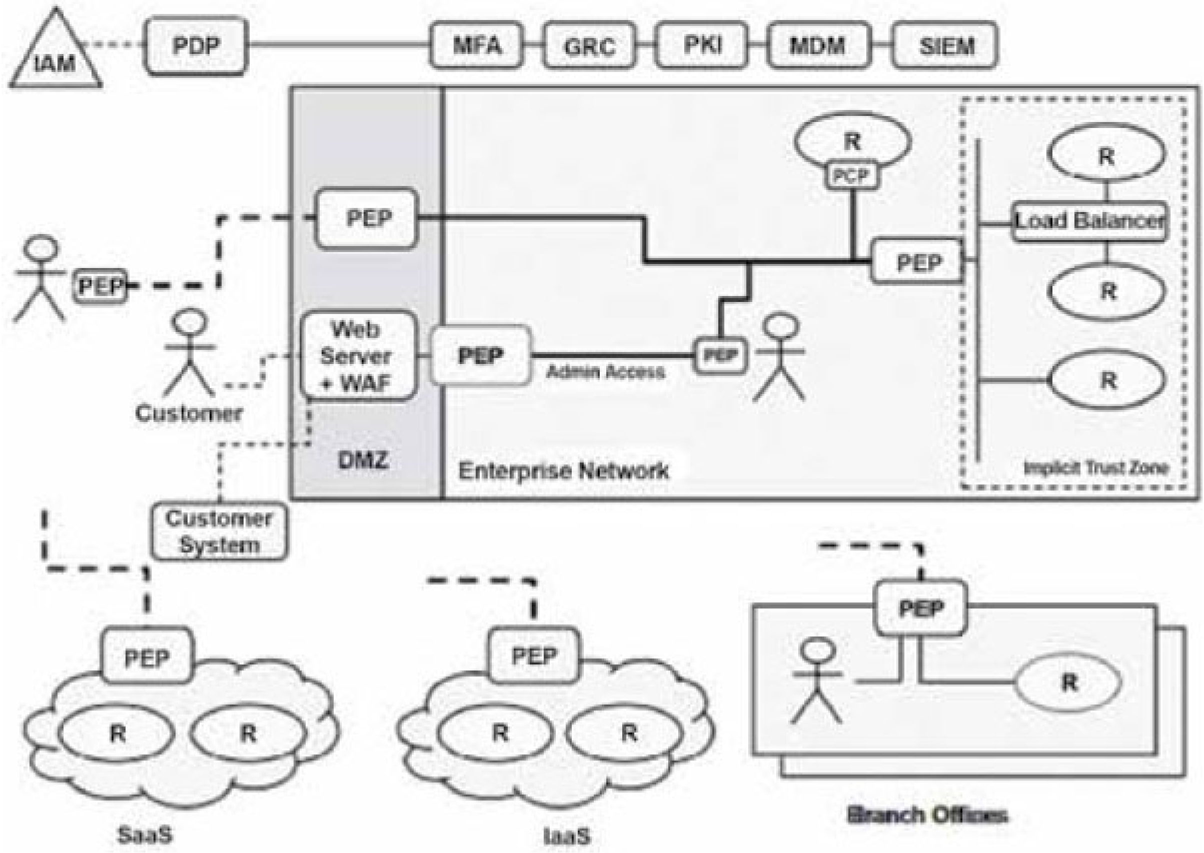


Figure 18.4: *Zero Trust Architecture*

The sample enterprise from Chapter 3: *Architectures With Zero Trust* has opted to install their "*full Zero Trust*" architecture in this diagram. They've integrated the majority of the strategies outlined throughout the book, resolving their worries and achieving their goals. Let's have a look at how they went about it. Of course, their PDP is linked to their business identity provider (IAM)—it's a basic need. Their PDP is also linked to other IT and security systems, such as their MFA, SIEM, GRC, endpoint management, and PKI systems. Throughout their infrastructure, they have a number of distributed PEPs, several of which are enforcing access to resource enclaves. On most users' devices, the organization uses local user agent PEPs, and it has also installed PEPs directly into select servers. Note that the PEP in the DMZ and the PEP in front of the implicit trust zone have an encrypted PEP-to-PEP connection—this is a setup allowed by various Zero Trust platforms. Their Zero Trust solution protects access to both SaaS and IaaS resources, and the PEPs in their IaaS environment make access control choices based on dynamic properties (metadata) on workloads. They've put the PEP at their branch offices in such a way that it handles resources and users in an IoT-style manner. That is, Zero Trust protected resources can be accessed (and accessed by) devices (and users) on that network. Finally, remember that not all parts on the network are in scope for the Zero Trust solution. In the IaaS system, for example, there are implicit trust zones (resource enclaves) between the resources in the corporate network. Also, while a PEP controls admin access to the DMZ web server, customer access to other services on that server falls outside the scope of the Zero Trust solution.

Considerations

Full Zero Trust is clearly a large endeavour, and even with top-down support and endorsement, it will be a technological and organizational problem. In truth, not all businesses will be prepared, especially as a first step toward Zero Trust. We'll go over this topic in greater depth in next chapter, but before, we'd like to make some suggestions.

Recommendations

While a large-scale network transformation project may not be feasible at first, we want to emphasise that limiting users' network rights is a critical aim; in fact, it's one of the most significant things you can do as part of your Zero Trust campaign. It can be done in stages, so even if you do it subnet by subnet (or VPC by VPC, or application by application), it will be beneficial.

We recognize that corporate networks are complicated, and that many in-place pieces may appear to be restrictions or impediments. This, however, does not have to be the case. Consider an office with printers that users have implicit access to when they enter the building. This access can simply be supplied by a Zero Trust policy; thus it shouldn't be a barrier to Zero Trust adoption. In reality, in-place components may be used to provide Zero Trust in some circumstances.

A NAC solution had previously been installed across 50+ branch offices for one of our business clients. They configured the NAC to assign users in the relevant groups to the guest VLAN instead of the employee VLAN as they rolled out their Zero Trust agent to users' devices, group by group, effectively taking them off the network. The beauty of this move is that end users were unaffected—they were able to stay completely productive while still having access to all of their programmes.

In some respects, each of the preceding six use cases represents a microcosm of the broader Zero Trust network scenario's concepts, tactics, and issues.

This is what makes this a fascinating set of difficulties, and it's also another solid reason to start with a smaller use case rather than a full-fledged Zero Trust implementation. You won't have to deliberately tackle every problem "at scale" if you start with a smaller scenario and user population, and you'll be learning and developing things (policy, teams, procedures, etc.) along the way, making it much simpler to achieve this bigger use case over time.

Conclusion

To summarize, we looked at seven distinct scenarios for implementing Zero Trust in the company in this chapter. The majority of these use cases have been addressed throughout the book, but this chapter allowed us the chance to analyze each one in greater detail, and to do so while building on the information and background we've gained over the previous 17 chapters. Take a breath and a step back as we emerge from the intricacies of these use cases—in next chapter, we'll look at how your business should approach Zero Trust from a programme and initiative standpoint to assure success.

Chapter 19

Creating a Successful Zero Trust Environment

We've covered a wide range of security and technical subjects in the first 18 chapters of this book, including Zero Trust concepts and architectural approaches, an examination of a large set of IT and security aspects, and a discussion of Zero Trust policies and use cases. Any discussion of Zero Trust must include those architectural concepts and technological aspects. However, one feature remains, as articulated in the most often asked Zero Trust question: "How do I get started?" That's a good question, but the question underlying it is what we feel is the missing topic: "How can I ensure the success of my Zero Trust project?" The goal of this chapter is to provide an answer to that question.

Our best one-sentence response is to take a targeted, gradual approach while maintaining sight of (and preparing for) your wider Zero Trust programme, and intentionally taking the time to develop bridges and channels of communication with your peers throughout the business. This isn't to suggest you can't have a standalone Zero Trust project, but Zero Trust by its very nature necessitates interaction with other IT or security components that will be owned or controlled by other teams. This involves communication and integration with those teams, which will be a big element in deciding how successful your Zero Trust initiatives will be.We'll look at this issue in depth in this chapter, giving you tips on how to get started and how to ensure that your project and the wider Zero Trust initiative are a success. Keep in mind that, like any large-scale business security or IT project, Zero Trust can present both technical and nontechnical issues. In fact, for technically oriented people like the authors of this book, the softer aspects of programme design, communication, and understanding organizational culture are sometimes more difficult than technology.

We'll look at Zero Trust projects from both a top-down and bottom-up standpoint. This is a simple and helpful method for us to categorise and discuss things, but it is an artificial distinction in reality. Every Zero Trust project and initiative will incorporate aspects of both approaches, so don't think of them as mutually exclusive—this is merely a convenient method to organise the topic in this chapter. Specifically, even if your organization has a strategic, top-down vision and mission for Zero Trust, you'll still have tactical projects and decisions to make.

Similarly, even a tactical "*under the radar*" Zero Trust project aimed at tackling a specific problem would include coordination and integration with other tools and teams, and so likely have at least some strategic aspects. In fact, adding strategic elements in a tactical first Zero Trust project is a great approach to set yourself up for successful second and third initiatives. After that, let's get started, starting with the strategic perspective.

A strategic approach to Zero Trust (top-down)

A strategic approach to Zero Trust (by definition) necessitates the presence of a champion at the executive level, ideally a C-level executive.

Because Zero Trust is not only an IT endeavour, cross-business-leader alignment is critical for complete endorsement and implementation of a Zero Trust strategy across the organization. While security teams may recognize that Zero Trust represents the state of the art in security best practises, that may not be enough compelling to stimulate the company to start on a purposeful \sZero Trust journey. A separate catalyst, such as new security or executive leadership, a data breach, M&A, or even a result of pandemic-driven access and security adjustments, may be required in many circumstances. Other catalysts could include changes in regulatory requirements or internal audit findings.Because this is a cross-organizational endeavour, security teams must be aware that there will be business objectives to meet and that this strategic plan will entail business and supervision processes. These should not be viewed as barriers, but rather as important vigilance in the execution of a business-critical undertaking. With this in mind, let’s look at different organizational structures that may be useful in a Zero Trust strategy.Keep in mind, not every organization has or requires all of these—we are to some degree showing the “maximal set,” which may only exist in larger and more formal organizations. Take stock of which of the following organizational structures are in existence or should be formed when you begin developing your Zero Trust approach. Governance Board, Architecture Review Board, and Change Management Board Let’s take a look at each one separately.

Board of directors

Typically, governance boards develop policies that provide the company direction and promote the organization’s overall (financial and human) health. Governance boards are frequently used to assist a business in achieving its Governance, Risk, and Compliance (GRC) goals, and may be part of a GRC group. They should include the following organizational aspects, since they will be crucial to Zero Trust:

- Risk
- Audit
- Operations
- Identity
- Security

The teams in charge of each of these areas should have a say in the Zero Trust initiative’s rules, and their advice and support will be crucial to its success. Specifically, when innovations are assessed and considered for inclusion in new projects, this board frequently has veto power.

Understanding and managing the organization’s risk threshold will be a critical factor in determining the amount of support for the Zero Trust programme at a higher business level.

Architecture review board

An Architecture Review Board (also known as a Corporate Architecture Board) is in charge of analyzing present and prospective enterprise technology and will be heavily involved in and relevant to a Zero Trust Strategy. The board of directors is also responsible for defining the enterprise's architecture standards, which are a critical component of any Zero Trust programme. Although the technological requirements for Zero Trust are difficult (as highlighted throughout this book), they may be swiftly integrated with current technology if architectural principles are followed.

Organizations have enterprise architecture boards for a variety of reasons, including uniformity and company-wide visibility. Finally, the members of this board will be able to share their combined expertise on the effects of environmental changes, which is clearly relevant to any Zero Trust programme.

Board of change management

Finally, each endeavor should include a Change Management board, since it will be responsible for the time and scheduling of introducing new solutions into a production environment. Application and infrastructure integration with Zero Trust technologies will become increasingly important as Zero Trust becomes a larger and more operational component of the company. Because integration and deployment can become more policy-based and automated with Zero Trust, this may actually speed up change management processes. Remember that not every business requires this degree of formality, but if you already have these teams and procedures in place, they will benefit your Zero Trust strategy and help you adopt Zero Trust aspects into your environment faster.

Drivers of value

While the deployment of Zero Trust is typically centered on technology, these initiatives will ultimately be driven by business objectives. Let's switch gears and talk about the business-level benefits that a Zero Trust effort may provide: Customer/Partner Integrations, Security, Audit and Compliance, Agility/New Business Initiatives, and Technology Modernization.

Security

Given that Zero Trust is focused on security, it's a natural value driver. As such, it's generally one of the driving motivations behind any Zero Trust programme. It's worth noting that the security advantage in any particular project might be as simple as implementing multifactor authentication into the user experience or as complicated as building an enterprise-wide Zero Trust network. Also, security may not be the primary priority for every project under a Zero Trust strategy in some situations. For example, you might be working on a project that uses a Zero Trust platform that has already been deployed to allow customers' systems to integrate with yours. This might not genuinely increase security, but it will meet the Customer/Partner Integration driver, which we'll talk about later.

Compliance and auditing

Audit and compliance enhancements may not be as evident or technical, but greater logging along with an identity-centric strategy will result in better audit results and compliance achievement. Meeting corporate auditing standards requires visibility into which identities are doing which business operations and accessing which technological assets. And, Zero Trust solutions frequently cut audit \scosts and cycle times, due to providing simply available and easily readable access records. Learn about the many types of audit log reports available in your Zero Trust system and how they correspond to the sorts of reports required by your internal and external auditors. This has the potential to provide significant value to your company.

New business initiatives/agility

Zero trust is frequently used to allow secure application agility or new business initiatives, both of which may be extremely beneficial to a company. Many firms, for example, are adopting a "*Cloud First*" strategy, and Zero Trust may be utilized to give guidance and guardrails. In general, Zero Trust's automated and context-based security approach is ideal for enabling fast-paced, creative business activities based on safe, accurate access restrictions.

Integrations with customers and partners

One of Zero Trust's fundamental ideas is to enable and profit from secure integration across traditionally isolated technologies. This is true both within and outside the company. As a result, businesses may leverage Zero Trust platforms to allow new sorts of customer and partner system, data, and process interactions. This might be as basic as allowing safe consumer access to a traditionally private online service or as complicated as real-time data sharing across businesses. Both have the potential to provide tremendous corporate value and innovation.

Modernization of technology

Finally, Technology Modernization is a broad value driver that may encompass a wide range of advantages, such as upgrading obsolete security or IT infrastructure, decommissioning now-ineffective technologies, and transitioning to contemporary alternatives. Although there will be others, most of this modernisation will be applied to the IT and security systems that we reviewed in Part II of this book. We've discovered that using these five broad categories to quantify and categorise the impact that each Zero Trust project will have as part of the enterprise's larger Zero Trust programme is a good approach to start. That is, it aids in the approximate quantification and visual representation of the response to the question "What is the business aiming to achieve with this time and money investment?" These value drivers will work for tactical and strategic initiatives alike (although the magnitude of benefit may differ). We'll walk through a sample case later in this chapter that shows how to represent these in a visual radar diagram. Through the course of your endeavour, these comparisons may assist you and your team objectively review, compare, and prioritise prospective initiatives. Let's take a practical look at how businesses should approach Zero Trust now that we've looked at it from a strategic standpoint.

A tactical approach to Zero Trust (bottom-up)

A tactical Zero Trust project is defined as one that has a limited scope and duration and is focused on tackling a specific set of challenges. Most significantly, the solution is addressed in a way that supports the concepts of Zero Trust security, and it may involve the organization’s use of new and alternative security tools and platforms. While a first Zero Trust project will bring new concepts and platforms (and hence change) to an organization, it must do so in a way that is consistent with the business’s broader security, risk, and architectural approaches. These sorts of independent ventures might come from a variety of places. They may, for example, be led by application teams with a specific access requirement. In this case, the security team may assist the app owners in understanding why Zero Trust is the ideal strategy. In fact, having a business or application group as a sponsor is a great way to get started with Zero Trust because they’ll be supportive of the project and can help you overcome any political or technical obstacles. In many situations, however, it will be the security team that pushes for a first Zero Trust project in order to tackle a specific security or risk concern with the goal of utilizing it to kick off their Zero Trust journey.These initiatives have a good chance of succeeding, but they run the danger of seeming to be a “solution in search of a problem,” and they may face resistance from networking or business teams that don’t see the benefit in change. Don’t disregard this risk or just hope that you don’t encounter it—this may be a real and major hindrance, as most Zero Trust implementations need modifications to IT elements beyond the security team’s scope, such as end-user experience, or network configuration. The goal is to find a prototype Zero Trust project that addresses some current pain problems for teams, particularly ones that go beyond security, and that will pique their interest and support for the initiative. These initiatives have a good chance of succeeding, but they run the danger of seeming to be a “solution in search of a problem,” and they may face resistance from networking or business teams that don’t see the benefit in change. Don’t disregard this risk or just hope that you don’t encounter it—this may be a real and major hindrance, as most Zero Trust implementations need modifications to IT elements beyond the security team’s scope, such as end-user experience, or network configuration. The goal is to find a prototype Zero Trust project that addresses some current pain problems for teams, particularly ones that go beyond security, and that will pique their interest and support for the initiative. Examine the six targeted use cases from Chapter 18: *Zero Trust Scenarios* to see whether any of them might make suitable initial projects. Keep an ear out for new business initiatives in your organization; if Zero Trust can make them more readily and securely possible, they could be a good match as well. Remember that significant Zero Trust projects must begin somewhere, and even in this case, we advocate starting with a smaller, more focused project for a variety of reasons. It provides you with a vehicle to do vendor or platform research as well as a smaller-scale **proof of concept** (**POC**). It also allows you to explore new things, make errors, and learn from them in a low-risk environment. Because every enterprise's IT and security landscape is different, every enterprise's path will be different as well. Accept this and learn by doing things over and over again. When you initially start, there will be a lot of unknowns, and you won't do everything perfect on the first try. The most essential thing is to show some progress and gain momentum and support for Zero Trust inside your company. Even the most tactical Zero Trust project teams will need to include individuals who are responsible for identity management and networking, as well as those who are responsible for enterprise architecture. The next section, which introduces a prototype bottom-up project, should assist clarify this.

Deployments using Zero Trust as an example

The purpose of this section is to demonstrate two sample Zero Trust deployments in terms of project and milestones. These should give you a good picture of how and why these two hypothetical project teams approached things the way they did. Remember that these are only example samples that are meant to assist you in making educated selections for your own projects.

Scenario 1: A Zero-Trust tactical project

In our first scenario, a transportation services company has outsourced financial management system operations to a third party, which supplies them with this vital business function through a pool of roughly 30 part-time financial analysts. The organization's finance systems remain hosted at headquarters, implemented on traditional hardware-based servers, despite the fact that the third-party users are distant (and in reality are situated in two separate countries). This architecture is required because the financial systems are linked to a number of other on-premises systems that are critical to the company's operations. Third-party users now access the financial system using a regular VPN, but the business now has various security issues that must be addressed owing to a change in IT auditors. They must now implement MFA for third-party users and hook into the identity lifecycle events of those businesses to guarantee that departed users' access is correctly revoked.While the security team could handle these issues by implementing a stand-alone MFA solution and creating a business process to assure user offboarding, they have been exploring and learning about Zero Trust and are happy to have discovered a focused beginning project. Figure 19-1 depicts a high-level perspective of the project timetable and flow, with actions split into those taken by the project team and those involving the enterprise architecture team. Note that in this case, third-party users already have access via the VPN solution, and the auditors are not seeking adjustments for another six months, so there is no pressing need to make a change right away.This is advantageous to the project team since it allows them to be more deliberate in their study and evaluation of Zero Trust platforms. With that background in mind, let's take a look at each phase of the project one-by-one.

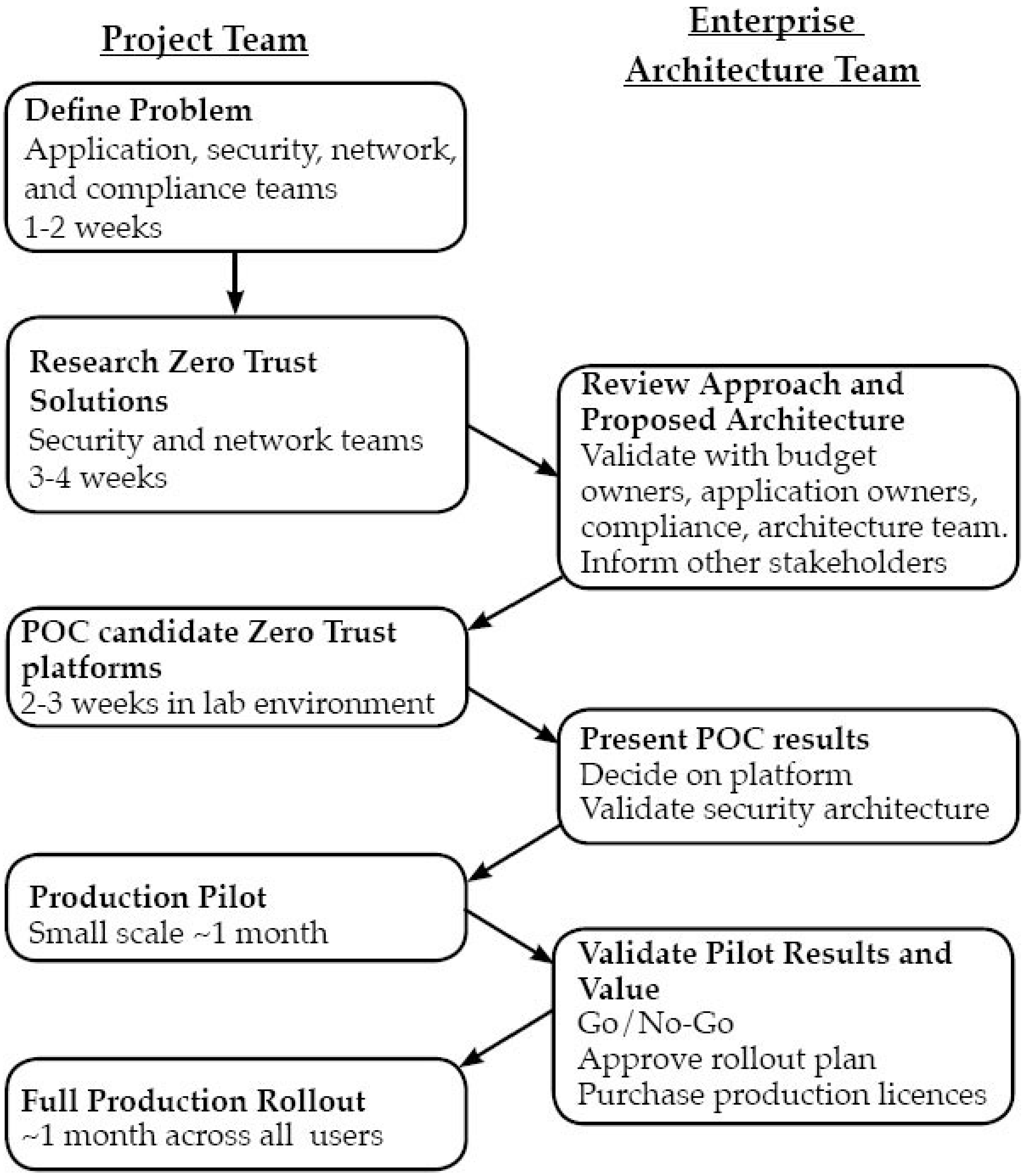

Project Team
Enterprise Architecture Team
Define Problem
Application, security, network, and compliance teams
1-2 weeks
Research Zero Trust Solutions
Security and network teams
3-4 weeks
Review Approach and Proposed Architecture
Validate with budget owners, application owners, compliance, architecture team.
Inform other stakeholders
POC candidate Zero Trust platforms
2-3 weeks in lab environment
Present POC results
Decide on platform
Validate security architecture
Production Pilot
Small scale ~1 month
Validate Pilot Results and Value
Go/No-Go
Approve rollout plan
Purchase production licences
Full Production Rollout
~1 month across all users

Figure 19.1: *Sample Tactical Zero Trust Project Timeline*

Define the issue

While the auditors only identified MFA and zombie accounts as issues that needed to be addressed, the security team also wants to implement additional security controls, such as performing basic device posture checks, geolocation checks, and ensuring that the user is in the correct directory group in the third-IAM party's system. The security team, which is in charge of this, devotes a few weeks of calendar time to educating the application, networking, and compliance teams on the project's planned scope as well as Zero Trust principles and goals.

Look into Zero Trust solutions

Following the stakeholder buy-in obtained in the previous stage, the security team spends a few weeks researching and evaluating Zero Trust platforms, examining a variety of products from major vendors, smaller vendors, and open source. The majority of the solutions are free to try, and technical members of the security team spend their leisure time delving into a single product and sharing what they've learned. Following this step, they choose two potential Zero Trust platforms and develop a security and deployment architecture.

Examine the approach and the architecture proposed

The team then presents the suggested architecture and project plan to the enterprise architecture team's key stakeholders, who include the owner of the financial application, compliance, network, operations, and the budget owner. Although this firm has a semiformal enterprise architecture team, the security team is choosing a more organized approach for this project, understanding that the scope and maturity of their Zero Trust programme will increase over time.

Candidate Zero Trust platforms in POC

The security team brings in their two candidate Zero Trust systems and does a Proof of Concept in their non-production lab environment once the enterprise architecture team has authorized the methodology. This allows them to compare these solutions to the given criteria in a quantitative way. Because this is a well-defined and straightforward case, it will only take them 2–3 weeks of part-time work to finish and select a platform.

POC results currently available

After that, the security team reconvenes with the enterprise architecture team to explain their results, showcase the higher-scoring solution, and offer a recommendation on the platform and security architecture. Integrations, user experience, and operational ramifications, as well as key security tasks, are all covered in this talk.

Production Pilot After the enterprise architecture team's stakeholders agree the strategy, the security team launches a Zero Trust platform pilot instance. They utilize this phase to communicate with the identity management team of the third party for integration and to roll out the Zero Trust (and integrated MFA) software to ten end users across the two sites. These consumers keep their old VPN connection on their devices so that if they run into problems with the Zero Trust strategy, they can quickly switch back without losing time. The security team takes roughly 1.5 weeks to put out the new system, and end users have another 2.5 weeks to utilize it in production. The pilot is basically a success, with a few small hitches and some user education concerns.

Validate and value the pilot results

The last official meeting with the enterprise architecture team is simple to arrange because the pilot was a success. The security team presents the findings and recommends a strong "*go*" decision, which is accepted. The team also gives their approval to their plan for bringing this to production (and, very importantly, decommissioning the current VPN solution). The team additionally buys production licenses from the vendor they've chosen.

Full-fledged production

For the remaining third-party users, the security team delivers the Zero Trust solution and decommissions their VPN access solution. They also utilize this opportunity to pass off the Zero Trust solution's production operations to their network operations team. This isn't surprising because this group has been involved throughout the process. Finally, while this was the first Zero Trust initiative, it will not be the last. The security team makes it a point to publicize the project's success, as well as the resolution of the outstanding audit problems, in order to build momentum and support for future initiatives based on their Zero Trust platform. A real-world project, of course, will be more complicated and require a lot more contact between the various teams. We're also using the enterprise architecture team as a placeholder; your company may have a team with a different name that serves the same purpose. Also, keep in mind that various companies have different approaches to things. For example, the enterprise architecture team may merely convene to be briefed by the security team in certain businesses, but in others, they may have decision-making authority (and, therefore, veto power over the project). Let's take a look at a completely different scenario, one in which Zero Trust is approached from a strategic standpoint.

Scenario 2: A Zero-Trust strategic initiative

This storey begins with a fortunate break. This pharmaceutical company's security team had recently recruited a junior security engineer, and one of their first jobs was to consolidate, reconcile, normalize, and assist the SOC make sense of the massive number of loud and dirty event logs coming from their hundreds of Windows machines. This is the type of unglamorous job that is frequently postponed in favor of more pressing chores. In one example, the engineer saw some unusual behavior that he reported as "*Hey, can someone assist me figure out what's going on?" " This does not appear to be correct*."

It became out that their network was infected with malware that was undertaking low-level reconnaissance. They swiftly enlisted the help of an outside Incident Response team, who effectively remedied the situation. The company understood how fortunate they had been during the post-event review. Although they assumed it was via a targeted phishing email, the malware's first entry point into the network was never completely discovered. They did, however, come to the conclusion that it was being managed by a distant command-and-control server and had been spreading through their flat network via a mix of unpatched Windows computers and weak admin password policies.

The IR team's main conclusion was that they appeared to have spotted the attacker early enough to avoid data exfiltration, but that if this had been a ransomware assault, the great bulk of their network would have been destroyed in a matter of hours. The strategic consequence of this "near miss” was immediate and decisive—as a pharma business, the confidentiality, integrity, and availability of their research data and production processes are critical to their success. The CEO empowered the CISO to make adjustments when the executive leadership team and Board of Directors insisted that these risks be addressed.

The CISO, whose security leadership team had been discussing and evaluating Zero Trust, devised a two-phase strategic plan for implementation. Phase 1 was designed to improve the security of the organization’s most valuable assets by requiring end users, developers, and system administrators to adopt Zero Trust access. The use of **multi-factor authentication** (**MFA**), deep device posture checks, enhanced network segmentation, and the removal of broad admin network access were among the new measures to be implemented. Phase 2 was supposed to further divide the network by using a Zero Trust café-style network to move all users “off net.” It also featured a move away from their sophisticated on-premises directories and toward cloud-based Identity-as-a-Service, which uses contemporary and passwordless authentication.

Finally, this phase was designed to embrace and expand the organization’s fledgling cloud-based IaaS and PaaS platforms, allowing for quicker and more effective cooperation with customers and partners.

Of course, each of these phases was broken down into its own project, which the company mapped out using the five value drivers:

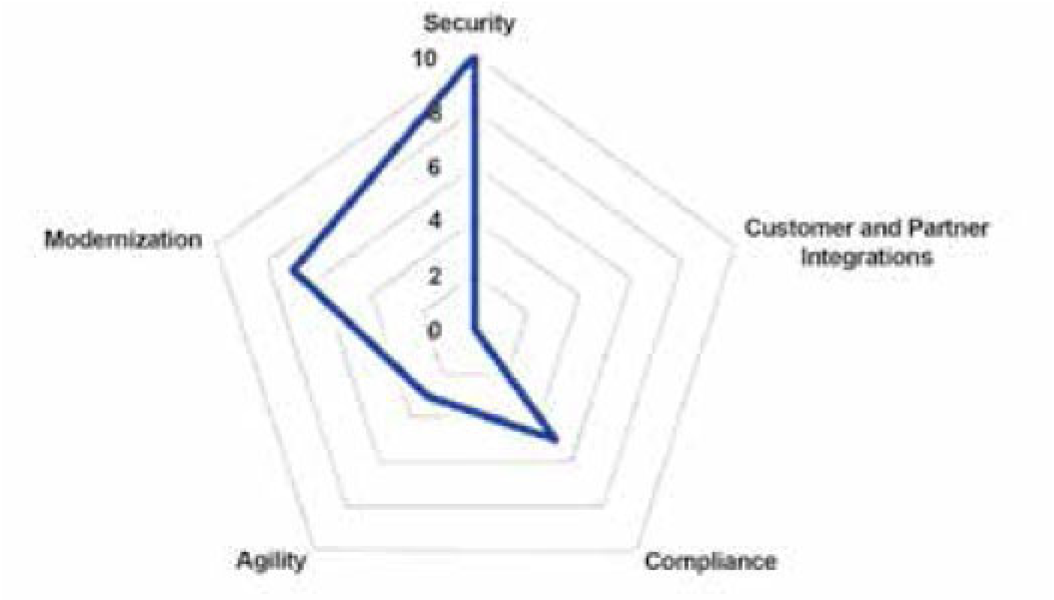


Figure 19.2: *Zero Trust Project Value—Radar Chart*

This project's initial goal was to improve end-user security for access to the company's most essential production systems. Before users were permitted access, the original set of Zero Trust rules needed MFA and verified device certificate and posture checks. They used the same restrictions whether the user's device was physically linked to the corporate network or not—after all, the virus that started the project was operating locally on the network. There was less of an influence on the other value drivers by design, in order to keep this initial Zero Trust initiative focused. This effort does resolve a number of open security \scompliance audit issues.However, it made no modifications to customer or partner integrations, and by removing numerous siloed access control systems, it only enhanced agility little. Given that this was their first production Zero Trust deployment, the team rated this project as significantly updating their security architecture. In conjunction with the initial project, the CISO and CIO partnered to create more formal structures and procedures around their existing Architecture and Change Management boards, ensuring that cross-team communication and cooperation were adequate. They opted against forming a formal Governance board since the Architecture board already factored risk and compliance into their decision-making. The CISO, on the other hand, decided to bring in an experienced outside consultant to assure neutrality and widen the team's viewpoint. Overall, with a strong catalyst and passionate CEO support, this example demonstrates how an organization may choose to execute the first element of a comprehensive Zero Trust effort. Of course, not every project will have the same amount of "juice" to free up funds, break down obstacles, and (if necessary) knock heads. The following section discusses some frequent roadblocks that security executives may face on their Zero Trust journeys.

Typical obstacles

Without a discussion of real-world issues related with Zero Trust projects and efforts, this chapter would be incomplete. IT and security in the enterprise is difficult and complex, and some Zero Trust projects will fail. This is regrettable, but true. The good news is that the majority of them will succeed, and the advice and suggestions we've offered throughout this book should put you on the right track. Also, keep in mind that any complicated system, including Zero Trust, will always have technological bugs, flaws, or rough sections. Perfection is an impossible goal, although significant advances in security and efficiency are feasible.After that, let’s look at some of the most typical hurdles and how to prevent or overcome them.

Immaturity in Identity Management

Zero Trust is inextricably linked to identity management, and Zero Trust initiatives may be hampered by a perceived or real lack of IAM maturity. The all-too-common “our directory is a mess” tale, the expansion of groups (often tens of thousands), or an ongoing endeavour to consolidate or reconcile identity providers are all examples of immaturity. Although this is the reality for many IAM teams, it should not be a deterrent to Zero Trust adoption.For user authentication, Zero Trust systems will utilize an identity provider, and you may determine how much IAM characteristics and groups to use in your Zero Trust policy. Because Zero Trust systems automate the use of certain identity features, even if just for a small slice of your IAM system, they may actually be a driver for greater maturity and data integrity. Remember that we spoke about this in Chapter 5: Identity and Access Management (IAM) in the section “Zero Trust as a Catalyst for Improving IAM.”

Political opposition

Unfortunately, some security executives may meet political resistance to reform in their companies. This is defined as those who erect hurdles to change despite the organization’s obvious benefits. Culture, technical prejudice, or an emotional attachment to present security techniques or systems may all play a role. There are a few options for dealing with this. Education comes first and foremost. Some individuals may be resistant due to a lack of knowledge, so educate them on the real-world benefits of Zero Trust and persuade them that it isn’t simply a marketing gimmick. Second, if your programme has a dedicated and enthusiastic executive sponsor, they should be able to overcome this obstacle. Third, you may recruit a project champion from within the line of business—projects that result in more revenues or reduced costs are particularly effective at breaking down barriers. Finally, this phase was designed to embrace and expand the organization’s fledgling cloud-based IaaS and PaaS platforms, allowing for quicker and more effective cooperation with customers and partners.

Of course, each of these phases was broken down into its own project, which the company mapped out using the five value drivers.

The *figure 19.2* depicts the initial project in this journey, which focused on fixing the most immediate security flaws found in the event, with the expected effect of each driver graded on a scale of 0 (low) to 10 (high) (high) *figure 19.2* depicts the journey, which was focused on fixing the most immediate security flaws discovered in the event, with the expected effect of each driver scored on a scale of 0 (low) to 10 (high) (high).

Constraints of regulation or compliance

Many businesses are regulated, or at the very least have data or systems that must comply with regulatory regulations. Government and industry regulations typically lag behind technology by a few years, making it more difficult for businesses to adopt modern techniques to fulfil these criteria. In many circumstances, your third-party/external auditor will be the deciding factor, therefore it’s critical to engage with them early. Don’t be afraid to collaborate with them and educate them early on in your Zero Trust project to ensure that they understand your goals. This will assist in achieving a great outcome.

Resource discovery and visibility

In a complicated company IT system, getting an accurate view of all resources can be difficult. This is especially evident in circumstances where there has been little monitoring or if things are moving quickly. This is frequently articulated in anecdotal statements like “I have no idea who is accessing what, how can I manage them?” Two distinct techniques are demonstrated in Chapter 4’s case studies. BeyondCorp and PagerDuty both used their Zero Trust systems to install fine-grained access control policies across complicated production networks. To guarantee that their technology didn’t disrupt user productivity, they used an observational technique, gathering and analyzing network data. This worked well, although it took time and effort. The Software-Defined Perimeter case study, on the other hand, adopted a step-by-step approach to enrolling individuals and groups. They also began with some more coarse-grained access controls and gradually tightened them over time. Both of these points of view are legitimate. It’s critical to remember that you get to choose where and how your Zero Trust platform is deployed, as well as how fine-grained the access restrictions are. So don’t make the mistake of presuming that complete visibility of every connection and data flow is required before you can begin. Use the data you already have or one of the numerous open source or commercial technologies available to enable network discovery and resource visibility.

The paralysis of analysis

The purpose of trying to thoroughly comprehend, identify hazards, and check out any new technology or strategy is admirable, but it has the unfortunate side effect of postponing any decision or action forever. This “analytical paralysis” is a source of frustration for all parties concerned. It may be organizational culture, or it could be self-imposed by a security team aiming to drive change via consensus, which is never an easy needle to thread. We’ve seen companies struggle with this when they start on Zero Trust initiatives, with programmes spanning years and getting only a few dozen customers in production. In retrospect (and in the abstract), this is obvious, but it is frequently difficult to perceive in the moment. This is because most of us, and most teams, want to plan, investigate, and validate our work properly and thoroughly. When businesses go on a purposeful Zero Trust path and must gain consent from a wide range of stakeholders before deploying anything into production, this form of paralysis can occur. This might be an issue. This is especially true if the other teams demand that the new systems satisfy the same operational maturity, automation, and integration standards as existing systems that have been in use for years. This can create a “*chicken and egg*” situation, especially if the project and design require the company to build a big and complicated infrastructure before ever deploying the first group of operational customers. We’re not suggesting that project teams or security architects take shortcuts or avoid conducting thorough research and validation. However, we recommend that teams cooperate with all necessary stakeholders and approach their effort from the standpoint of getting Zero Trust into pilot or production as soon as feasible, even if the scope is initially constrained. While operations teams are naturally cautious about change, the majority will be eager to work with you.

For example, until the team has a high level of trust in the new system, you may recommend operating Zero Trust access in parallel with old access methods. The outdated access mechanism would subsequently be decommissioned. We don't want to conclude on a negative tone as we wrap off this section on typical barriers. Zero Trust initiatives, like all business IT and security projects, are fraught with risk and unknowns. Even if they encounter a few speed bumps along the road, the great majority of well-run initiatives are successful and offer value to the business. The most essential thing, in our opinion, is to iterate, learn, and not be scared to make modifications to your Zero Trust architecture in process. Ascertain that each Zero Trust project is broken down into manageable, attainable milestones. You'll never know all the answers at the outset of your journey but do enough homework to know some of the answers and the majority of the questions. Have trust in yourself and your team; you'll figure it out as you go.

Conclusion

We discussed top-down and bottom-up methods to Zero Trust in this chapter. In fact, most businesses will combine components of both in a hybrid strategy. In all circumstances, we feel that choosing a suitable first candidate project is critical to success. To acquire ideas about where to begin, look at the six targeted use cases from *Chapter 18: Zero Trust Scenarios*. Also, start sharing the ideas behind Zero Trust and the benefits it may give with your colleagues throughout your business, and ask a lot of questions. Are there any areas in which the company is currently experiencing operational, security, efficiency, or user experience issues? Is there anything from the audit that needs to be addressed? What about initiatives that use innovative environments like IaaS or PaaS? Are there any issues with a low risk yet great reward? Consider if you want your first endeavor to be high-profile or low-profile. There's no incorrect answer! A lower-profile project allows you to make errors (and learn from them) with fewer ramifications, but it also means you may have to compete harder for resources. A higher-profile project can break down these barriers, but it may also attract more scrutiny and a lower tolerance for errors. The strongest evidence of Zero Trust project number 1 success, in our opinion, is that you can instantly gain passionate support for project numbers 2 and 3. Be aware of the sorts of qualitative and quantitative metrics that matter to your company and be ready to record and deliver them in order to illustrate the value you've gained. Build relationships with your coworkers at all levels of the company. Both strategic and tactical Zero Trust programmes need enterprise-wide change, which can be difficult to execute without assistance. Zero Trust programmes might be tough, but the outcomes are worth the effort.