CHAPTER 18

Zero Trust Scenarios

Throughout this book, we’ve examined many different aspects of enterprise security and IT infrastructure. We’ve looked at things from a technical and architectural perspective, and have mentioned various use cases throughout. In this chapter, we’ll be examining seven different scenarios, and discussing how you can evaluate and approach them for inclusion in your Zero Trust program. This is not an exhaustive set of use cases, but does cover most of the major scenarios.

Our goals for this chapter are to arm you with an understanding of how and when these different scenarios would be applicable in your environment, and to provide you with relevant recommendations for how to approach them. Of course, these scenarios also need to be looked at from a deployment and operational perspective, which will be part of our discussion in Chapter 19. Finally, for the sake of brevity, we’re not going to be spending very much time here justifying these scenarios—hopefully, if you’ve made it all the way to Chapter 18, we’ve already convinced you of that. Let’s dive in, starting with one of the most common Zero Trust use cases, which is replacing a VPN.

VPN Replacement/VPN Alternative

We talked about VPNs, their weaknesses, and the comparative benefits that Zero Trust provides previously, in Chapter 9. In this section, we’ll briefly reiterate this use case in order to frame up a discussion about how you should approach a Zero Trust project focused on an enterprise VPN (remote user access) use case. Note that we’re examining two related scenarios:

- Replacing an existing, in-use VPN with a Zero Trust solution
- Deploying Zero Trust for a new remote access scenario

While these two scenarios have similar technical considerations, they should clearly be approached from different perspectives in terms of justification and decision-making. New projects often represent simpler and easier decisions, as there won’t be as many constraints
or dependencies in place. This contrasts with a VPN replacement scenario, where there’ll need to be a justification for replacing an in-place and operational VPN solution. This is not to say that this is a significant barrier—we’ve seen many, many VPN replacement projects—just that security leaders need to be prepared to discuss and justify the decision and project from potentially several perspectives, including security, technical, operational, and financial. Note that we do strongly recommend that organizations replace their VPNs with a Zero Trust approach; there are many good reasons to do so.

Let’s briefly review the architectural differences between traditional VPNs and a Zero Trust model, introduced in Chapter 9, and consolidated in Figure 18-1. Note that this scenario is only focused on providing remote users with secure access to services.

**Figure 18-1. Enterprise VPN and Zero Trust Architectures**

240
Traditional VPNs can only establish a single secure network tunnel from the user’s device to a VPN server, which terminates the secure tunnel, and permits network traffic to proceed into the private network area. VPNs perpetuate a perimeter-based network model, requiring that any distributed resources be connected to the enterprise’s core network over a WAN. Alternatively, they will require users to manually switch VPN connections when they need to access resources in different locations. In contrast, Zero Trust systems will establish multiple secure connections to distributed PEPs, so that users can access them transparently. (Note that this is true for the cloud-routed and enclave-based models. It may not necessarily be true for the microsegmentation or resource-based models, depending the specifics of the implementation.)

**Considerations**

In this section, we’ll look at a few different angles, to help you identify candidate VPN projects for Zero Trust.

**Resources**

Look at the number, type, location, and value of the resources under consideration. How business critical are they? If this is a replacement, how are they being accessed today, and what headaches or pain points are associated with the current VPN?

Generally, Zero Trust solutions provide better performance than VPNs, especially for distributed resources. They also can often be deployed to protect resources in locations or environments where the enterprise cannot deploy a VPN entry point, for example, on a third-party network. If you have a highly distributed or highly dynamic set of resources, these will likely be good candidates for a Zero Trust approach—recall the dynamic target rendering from our policy model chapter.

**Users and User Experience**

Who are the users currently utilizing the VPN, or who need to access these new resources? Are the users all remote? Was this remote user access solution deployed rapidly (and potentially with some known issues or compromises), for example, in response to the COVID-19 work-from-home shift? Are on-premises users accessing these resources through a separate security model—for example, via firewall ACLs?
In these cases, there are often good reasons to adopt Zero Trust, for example, to overcome security or operational issues caused by a rapidly deployed VPN. If there are resources that were deployed recently, it may be the case that only remote VPN users have a secure access pathway and that your organization needs a solution for on-premises users. And finally, Zero Trust solutions, designed to secure access for all users to all resources, can eliminate siloed solutions, such as separate rules and access mechanisms for remote vs. on-premises users.

Zero Trust can be applied incrementally, group by group, or application by application, although end-user experience should definitely be a consideration. That is, be aware of the different access tools that your initial sets of users utilize, to avoid imposing unnecessary friction. For example, you likely shouldn’t require that a set of end users switch back and forth between their current VPN and your Zero Trust solution throughout their workday. It’d be far better to have a group of users switch over to your Zero Trust solution for all their access needs, combining their current broad VPN-level access with more precise Zero Trust policies for specific resources. This way, they begin to obtain improved security while also obtaining an improved user experience. We’ll be discussing this further in Chapter 19.

Identity Providers

Some VPN implementations are not integrated with enterprise identity providers; in these cases, a Zero Trust deployment can quickly deliver considerable value. By tying remote access user authentication to their enterprise identity provider, security teams eliminate an identity silo that existed within their VPN. This eliminates any work necessary to keep that silo in sync with their primary provider, for example, to respond to identity lifecycle events of Join, Move, and Leave. Even if a VPN uses an enterprise IdP, a Zero Trust solution will improve on it, by enforcing fine-grained and context-sensitive access policies. Many Zero Trust solutions also support multiple identity providers of different types, so that different user groups can authenticate against different IdPs, or so that legacy systems can be protected by modern authentication protocols.

Networking

It’s critical that you obtain a clear understanding of your enterprise’s network topology, data flows, and where the protected resources are housed. This knowledge will enable you to make well-informed decisions and recommendations about transitioning
access from VPN to Zero Trust. Start by asking where VPN concentrators (entry points) are located, which networks they grant access to, and how distributed resources are accessed from a network perspective.

As we mentioned in the introductory part of this chapter, determine whether users are just accessing resources via a single entry point into an enterprise network. Even in this simple case, Zero Trust can add value, such as improved performance and stability, better integration with identity providers and MFA, and, of course, fine-grained access controls.

Teams or projects that require access to distributed resources typically struggle with VPNs, and is a situation where Zero Trust shines (as long as your chosen implementation and deployment model supports multiple concurrent connections to distributed PEPs). View this as an opportunity to ask “what if” questions of the networking or application teams. “What if users could access both these resources simultaneously?” “What would it mean if we could tie access to a business process, such as a service desk ticket?” “What if we could perform deeper device posture checks before users are permitted access?” These are excellent questions to spark conversations with those teams and get them onboard as supporters of your Zero Trust project.

There are other questions to ask your networking team as well, which will help you better plan for and advocate for your Zero Trust implementation. For example, find out what types of remote access policies (ACLs) your current VPN implements. How broad or narrow are they? If they grant very broad network access, which is quite common, your Zero Trust project can deliver improved security and reduced risk by greatly reducing network access without sacrificing user productivity. Determine whether there are any outstanding compliance issues or audit findings that your project can address.

If your VPN does impose restrictive network access, find out how well that works from an operational and user productivity perspective. It’s likely that this causes operational effort as well as user friction in all but the most static environments. Your Zero Trust solution should be able to impose equally tight (if not tighter) access control restrictions via automated policies, relieving your IT and operational teams of manual effort.

Finally, learn how your organization is utilizing any wide area networks. These typically impose considerable costs on organizations, and Zero Trust solutions can reduce (and, in some situations, eliminate) WAN usage.
Recommendations

VPN replacement and VPN alternative are a common first Zero Trust project, and are often a good one to get started with. The benefits are clear, and the functionality of a traditional VPN is generally quite easy for a Zero Trust solution to replace. We do recommend an incremental deployment, with consideration for those user groups who may need to retain both Zero Trust and VPN access for a period of time. These solutions generally can co-exist in harmony on an end-user device, but they generally cannot be running at the same time, as they’ll conflict at the networking level. This may be a concern if you have an “always on” VPN, for example, or want to deploy Zero Trust in a similar type of model.

One final recommendation for the VPN replacement scenario is to look carefully at the set of tools and processes that have been built around the scope and functionality of the VPN tool. Some organizations, especially those with older VPNs and older infrastructure, may have built a “web” of interdependent tools. This can pose a complex impediment to an incremental Zero Trust rollout. For example, one enterprise we worked with had a traditional VPN, which logged certain events into the user’s Windows event log. They’d built a set of “glue” tools which watched the Windows event log and responded to those events by performing some network configuration tasks. Modifying these tools was an additional task, and imposed a delay on the project, as that component was maintained and managed by another team within the organization. So, be cognizant of how your enterprise IT environment operates, and ask a lot of questions up and down the IT stack, and across your IT and business process ecosystem. You may be surprised at areas and ways in which the organization has built dependencies on specific tools or workflows. Some of these may be barriers to Zero Trust adoption, but some may be current pain points that your project can eliminate. There is often a considerable set of headaches around VPNs, which again is why they often represent a sound first Zero Trust project.

Third-Party Access

Third-Party access is also a good candidate scenario for Zero Trust, since it’s typically a source of headaches and risk for enterprises, and there’s a clear distinction and benefit from taking a Zero Trust approach vs. traditional third-party remote access. Let’s begin with a definition—for our discussion here, a third party is a non-employee individual with whom the enterprise has a legal relationship and who needs legitimate access to the enterprise’s network and private resources. Specifically
• The individuals can be identified.
• The resources they need access to are known and identifiable.
• They require access to private company resources (if all they need is internet access, they could just use the guest network when on-premises).

Note that we’re excluding full-time contract (non-employee) workers from this scenario; in our experience, these folks are treated much more like regular, full-time employees from an IT perspective. That is, a contract programmer on a 6-month assignment may not be a company employee, but will typically be issued a company-managed device, and be part of the enterprise’s identity management system. From a security perspective, they should be managed in the same way as employees, albeit with much more restricted network access.

Let’s look at a few examples of the type of third parties that are relevant to this scenario. These are often tied to outside firms with specialized expertise in a specific area, which doesn’t make sense to have internal to the enterprise. For example, one classic third-party access risk is a firm responsible for the monitoring, maintenance, and servicing of building HVAC systems. These systems are typically on the enterprise network, and HVAC vendors require periodic access to those systems to keep them running efficiently. Another example is a firm that has outside financial auditors who need access to an on-premises financial management system.

These types of third-party users are exactly the ones who require additional security controls. As the NIST Zero Trust document states, “An organization cannot impose internal policies on external actors (e.g., customers or general internet users) but may be able to implement some Zero Trust-based policies on nonenterprise users who have a special relationship with the organization”.

Our Zero Trust principles require that these users be authenticated and their network access be restricted to the minimum possible. Traditionally, organizations have used VPNs to provide remote access for third parties, and of course, VPNs exhibit all their weaknesses for third-party access. In addition, these third-party users are not employees, so by definition, they’re not using devices that are managed by that enterprise. This means that the enterprise cannot mandate or rely on the security posture of that device, which makes it even more important to impose security controls around the network access for that device.
One final constraint is that security teams cannot in general require the installation of any specific software on those devices. This is a little less absolute than it used to be, especially with the growing prevalence of Bring Your Own Device (BYOD) and the acceptance of using personal mobile phones or tablets for work activities. For example, a third-party user may be unable to install remote access software on an enterprise-managed laptop, even if that access is a required part of their job. But it’s becoming more acceptable to install remote access software on a personal tablet or a BYOD-device and use that for those work tasks.

Even if the third-party user can install remote access software on their device, it’s very unlikely that they’ll abide by the installation of more invasive endpoint management or security software, and it’s not realistic to include these third-party devices in your enterprise’s security or IT management system. Organizations simply need to accept that these systems and devices may not meet their security standards and use Zero Trust to enforce the principle of least privilege, as well as MFA. We’ll talk more about this shortly, in the “Recommendations” section.

Considerations

Third-party access in general is a good candidate for a Zero Trust project, and can sometimes serve as a useful first such project. These users tend to be very well-defined, and their access is typically limited to a small and static set of resources. They also typically represent an area of risk, since these users are accessing enterprise-managed resources from devices not managed by the enterprise.

Architecture

Third-party access network architecture is likely going to be similar to your VPN; in fact, it’s quite likely that these people will use your existing enterprise VPN. What’s important to understand is, like in the VPN use case, how and where these people are getting onto the network, and how their network traffic traverses the enterprise to reach their target resources. The type and location of these resources should affect the placement of your PEPs, and allow you to avoid having third-party user traffic transit very much of your network. As always, the principle of least privilege applies here, and your PEPs should prevent all unnecessary network access for these users.
Users and User Experience

User experience can be a less important consideration for third-party users, compared with employees. This is especially true if this access is only required intermittently rather than daily or on an all-day basis. For example, transparent (always-on) access to Zero Trust–protected resources may be desired for employees, but not necessary for third-party users. Having said that, clearly you shouldn’t deliberately make their access difficult.

Zero Trust systems often support both agent-based and agentless access, and third-party access is a use case where agentless access is often required. Depending on the type of resources being accessed, and the network protocols being used, agentless access may be a viable option. Typically, web-based applications are easily reachable with an agentless model, while non-web (non-HTTP) applications can pose some challenges. If a Zero Trust agent is technically required on user devices, but the third party refuses to install them, there are some alternatives, albeit at an additional cost. For example, the enterprise could host a virtual desktop for third-party users, into which they’d install the Zero Trust agent. Or, the enterprise could provision a managed device to be used by the third parties exclusively for access into the Zero Trust-protected environment.

Recommendations

From a user authentication and identity management perspective, we recommend having your Zero Trust system use the third party’s enterprise identity management system for authentication, if possible, but only if you have a sufficient level of confidence in their maturity and identity lifecycle processes. If not, have them utilize an IdP under your control—either your primary enterprise IdP or a smaller and simpler one dedicated to third parties. Any Zero Trust solution should be able to support authenticating different user populations against different IdPs.

We also recommend that you enforce MFA for these users, each time they attempt to access your resources. This form of step-up authentication should be implemented using an MFA provider under your control, and integrated with your Zero Trust system. This ensures that you can enforce your security policies regarding the frequency and type of authentication, and eliminate the potential for credential sharing by third-party users (which is a common occurrence).
You should definitely have your Zero Trust system enforce contextual access controls, such as geolocation, and configure fine-grained access policies that restrict user access to the bare minimum. These policies should be straightforward to define, since third-party access is typically only granted for a fixed and well-defined set of targets. We also recommend that you consider tying your third-party access policies to a business process when possible, in order to further restrict (and document) this access. For example, many Zero Trust systems permit the creation of policies in which access is controlled by the existence and state of a service desk ticket. This approach will work well in scenarios where third parties only need periodic access, ensuring that all access is requested, approved, and granted only for a limited period of time.

Finally, note that if an enterprise has already made the transition to Zero Trust, and has a “café-style” network, that even on-premises third-party users must access resources from within the Zero Trust model. That is, any third-party users who are physically present in an enterprise facility will automatically only obtain the same limited access they receive as when remote. This is an important benefit of Zero Trust—occasional in-person network access by third parties no longer puts the full enterprise network at risk.

Cloud Migration

Migrating applications and functions to cloud platforms is without a doubt a huge part of today’s enterprise IT and application development, and encompasses a wide variety of scenarios. The power of these platforms and the ubiquity and reliability of network connectivity make this a basically unstoppable trend, which is why it’s important that Zero Trust projects and leaders embrace this and educate their colleagues on the business and application development side about this new approach. Ideally, security teams will have in place a Zero Trust platform and a structured menu of approaches and approved components, which will enable application owners to quickly embrace the cloud.

Migration Categories

Of course, “cloud migration” isn’t one thing, it’s many different types of things, depending on many factors. But, in general, we believe these migration projects fall into four categories.
Forklift Migration

In this scenario, the application is moved from an on-premises physical or virtual environment to an IaaS environment “as is.” That is, there are no changes to application logic, topology, or technology. The end result is that the same application is running in a different place. Because this maintains the application’s structure and interdependencies, this migration can be faster and simpler, but delivers more limited benefits. This migration doesn’t require any development changes to the application; it should only require reconfiguration, and is well suited to COTS applications that the enterprise has licensed and therefore cannot modify.

Refactor the Application

In this scenario, the application is migrated to an IaaS environment, but it includes some technical or structural changes, ideally to take advantage of its new cloud platform. For example, the application may be modified to use a cloud-native database, or a cloud-based identity provider. Or, some of the deployment or operations infrastructure within the applications (such as the web server, or a logging server) may be rehosted on a cloud-based variant. This migration requires technical or development changes to the application, and can typically return moderate improvements. Some COTS applications will support this migration in some minor ways, for example, by supporting the use of a cloud-based database.

Rewrite the Application

This approach is the most technically difficult, but potentially provides a tremendous amount of value. In this model, application developers have the opportunity to completely rethink the application architecture, including taking a “radical” approach to embrace modern components such as containers, PaaS, microservices, or NoSQL databases, among others. Depending on the current application architecture, developers may be able to reuse elements of the application logic and data model to help accelerate things. This approach is not applicable to COTS applications.

Adopt SaaS

With this approach, organizations are making the shift from on-prem applications (either custom or COTS) to a cloud-based SaaS application. This, of course, represents a wholesale shift in application topology and access controls. It may be possible to reuse
some of the on-premises application logic, especially if the enterprise is adopting the SaaS version of their on-premises application. Enterprises should be able to import some of their application data, in order to jumpstart their SaaS application’s value.

In general, many (if not most) cloud migration projects are prime candidates for Zero Trust, because they encompass changes to security, network, and architecture, and therefore present an opportunity to embrace a modern and cloud-friendly security platform. In particular, Zero Trust systems, by their nature of being dynamic and context sensitive, can take advantage of the rich set of APIs presented by cloud platforms.

**Considerations**

These four migration scenarios each represent different opportunities to apply Zero Trust, which can definitely bring value and improve security for these in-motion applications. Let’s look at these from an architectural perspective.

**Architecture**

As we look at these scenarios, reflect back on the discussions in our chapters on IaaS and PaaS, and SaaS, where we talked about the network access controls and architectures associated with those models. Look at your organization’s planned or in-progress cloud migration architecture and approach, and influence them to ensure that they’ll work most effectively with your chosen Zero Trust network topology and access policies. And pose the following questions to yourself and your organization, based on your chosen cloud migration approach.

**Forklift**

Is the application self-contained, and are all parts being forklifted up to the cloud? Most applications are not 100% self-contained, so if this is the case, how are data flows in and out going to be managed? How can your Zero Trust PEPs facilitate this? Are all the (non-user) components of the application going to reside within an implicit trust zone? If so, is that risk acceptable with your new security model? If not, how will they be authenticated and obtain access across via a PEP?
Refactor the Application

In addition to the preceding Forklift questions, what is the current and intended network topology? What is changing about the component interactions? In what ways can you influence the changes to the application design?

Rewrite the Application

To what degree is the application team going to be “starting from scratch,” as they create a new application architecture? How will existing application components (either functional or data) be carried forward? Can the new architecture be aligned with your Zero Trust platform? Will the old and new versions need to coexist for a period of time? If so, will they need to exchange data? How will that be secured? Finally, can the application be written in a forward-thinking fashion to consume Zero Trust policies from the PDP and become an application PEP?

Adopt SaaS

This is clearly a different approach than the previous three, since the new platform is not under control of your enterprise. This may be simpler migration from a security and network perspective, since the destination is fixed. But definitely examine the SaaS platform from a security perspective and, using the guidelines we introduced in our earlier chapter, determine whether it makes sense to apply Zero Trust security to this SaaS environment.

Users and User Experience

In most cases, these newly migrated applications will have different network access models as a result of their migration to the cloud. This can disrupt or challenge the end-user experience. Your Zero Trust solution can often eliminate this friction, giving users transparent and secure access to these cloud-based applications while also enforcing dynamic and context-sensitive access policies.

Recommendations

We wholeheartedly recommend that as these applications migrate into a cloud environment, you collaborate with the application owners and include Zero Trust as part of the migration and deployment plan. The only exception to this may be adoption of SaaS applications, which may not need Zero Trust in every environment.
Finally, be proactive and collaborate with your application owner colleagues. Exposing them to your Zero Trust platform architecture and roadmap can in fact be a catalyst for accelerating cloud migration projects.

**Service-to-Service Access**

Service-to-service access control is definitely a legitimate, valuable, and important Zero Trust use case. Still, many enterprise Zero Trust implementations start with and focus on user-to-service access, for good reasons. Users and servers live in very different worlds, and have very different risk profiles.

**Users**
- Are untrusted and unpredictable
- Run their devices on untrusted, unmanaged networks
- Are mobile—access from different and changing locations
- Tend to lose their devices
- Often reuse passwords, or choose poor passwords
- Visit essentially random Internet destinations, and cannot operate with a whitelist of Internet destinations without impacting user productivity
- Receive email with phishing links, and occasionally click them
- Install arbitrary and unmanaged software on devices

That is, users are unpredictable, creative, and error-prone human beings. On the other hand, servers (and the services that run within them) are, or at least should be, the polar opposite:
- Run on enterprise-managed networks.
- Are more trusted—100% of the services running on any given server should be known, managed, and controlled by IT.
- Don’t visit random Internet destinations—in theory, the set of internal and external network destinations can be known and whitelisted.
- Don’t receive email with phishing links.
- Don’t lose themselves in bars or restaurants.
In fact, servers are trusted enough that many Zero Trust architectures include a segment of the network, behind a PEP, where servers communicate outside the control of the Zero Trust environment—the implicit trust zone, which we’ve discussed throughout the book.

Now, to be clear, we’re not attempting to dissuade you from applying Zero Trust to a service-to-service use case; we’re just highlighting that user-to-service often represents a higher risk. Nevertheless, service-to-service access controls should be part of every Zero Trust initiative, and may even make sense as one of the initial use cases. Let’s review the value and benefits that Zero Trust can bring to this scenario.

Most importantly, Zero Trust enforces the principle of least privilege, which is key to reducing the attack surface and reducing the blast radius of any successful attack. This brings with it an associated reduction in risk. It also ensures that, because all communications are explicitly granted by policies, there’s “top-down” visibility and control of service-to-service communications. That is, security and networking teams no longer have to rely on detecting communications occurring between services over a given protocol. Instead, the Zero Trust system, because it operates on a default-deny basis, ensures that all service-to-service communication occurs if and only if it’s been granted by a policy, and therefore is explicitly allowed.

This has an interesting effect—it actually serves as a form of referential integrity for the network—because all service-to-service communications must be permitted by a granted policy, it ensures that this communication is anticipated by deployment systems and processes. Because unexpected communications pathways will be blocked, it helps improve the maturity and predictability of the development and deployment process. While this may appear to impose additional friction, it’ll be more than repaid in terms of increased reliability, ability to automate, and improved security and resiliency. And, it ensures that deployed services are documented, and cataloged, eliminating the “don’t touch that server, we don’t know what it does” problem.

While this may well seem sufficiently valuable to justify the service-to-service use case, it also brings additional benefits. Zero Trust brings an overall reduction in risk, and an associated improvement in compliance. There are many compliance-driven controls that require better network segmentation, especially for high-value workloads. Zero Trust also ensures that network traffic is encrypted, in case applications are using unencrypted protocols. And finally, the fact that Zero Trust systems can dynamically and automatically respond to changes within the set of protected resources means that enterprises can adopt high-velocity development processes (such as DevOps, which we discuss shortly) without sacrificing security.
Considerations

Looking at Zero Trust models in the context of service-to-service, microsegmentation seems to be the obvious choice, and it may well be the best fit for environments where all servers have identities, and can be authenticated. This is a necessity, because recall that in the microsegmentation model, all servers are identities (Zero Trust subjects), and the access control mechanisms tend to reflect this service-to-service symmetry.

The enclave-based and cloud-routed models will also work for this use case, and in fact may be a better choice for environments where you’re just getting started with Zero Trust. These models give you more flexibility, especially when you have an environment where some identified and authenticated services (subjects) need to access remote services which are targets that are protected by a PEP, but are not themselves Zero Trust subjects. In fact, this is likely going to be a common server-to-server scenario in many deployments—asymmetric service-to-service, where one service is an authenticated identity, and the other service is not, but sits behind a PEP, as shown in Figure 18-2.

![Figure 18-2. Asymmetric Service-to-Service](image)

This model is a good alternative to “pure” microsegmentation that requires every service to be an identity, which may not be a good fit for some organizations or architectures. This approach is also useful for securing service-to-service access across different networks, especially for distributed application components that may occur as a result of a cloud migration. Cross-network service-to-service access control is a good use case for Zero Trust, since there’s inherently a need for a security overlay that normalizes the access control model in use.

Actually, there is one additional service-to-service approach that we need to mention, which is using an IoT-style non-identity access control method. As we talked about in Chapter 16, in this model, neither of the services are authenticated identities. That is, you can decide to treat your connection-initiating services as if they were an IoT device, with access controls based on weaker forms of identification.
and authentication, such as MAC address, IP address, VLAN, or switch port. This is possible, but has some downsides, as we discussed in Chapter 16. For these reasons, we don’t recommend this approach for service-to-service, if possible—it’s far better to authenticate at least one of the identities.

**Recommendations**

One way to identify good candidates for the service-to-service use case is to determine where you have servers that are communicating across network or domain boundaries. This will be a natural place to deploy a PEP, because the traffic is transiting a network boundary. As a result, this can be a relatively simple problem to solve.

Targeting peer servers on a single internal LAN may be more difficult, depending on the network configuration and on how difficult or easy it is to isolate the servers behind a PEP. On the other hand, high-value or compliance-driven server isolation can be a good reason to prioritize this scenario, especially if there is a strong need from a risk or audit finding perspective. These drivers can be a catalyst for making the necessary network and access changes.

As you consider this use case, look at your environment and try to identify services that would be a good fit—in particular, services that are high value, well understood and well controlled, and perhaps highly dynamic and difficult to secure with current solutions. Automated Zero Trust policies can be a big help here, adapting access to mirror changes in your server environment, without requiring manual effort.

Also recall that many servers host multiple services, and you can choose to place only some of the services behind a PEP, leaving the others unchanged. For example, you can deploy a PEP to control server-to-server access for a Database service running on a given host while still permitting non-Zero Trust users to directly access a web server on that same host.

Finally, take a look at any microservices environments your organization has deployed. As we discussed in Chapter 14, a microservices environment such as a service mesh may not be the best Zero Trust candidate, since it likely has its own internal and self-contained authorization model. But service-to-microservice can be a good place to start, as long as there’s a clear demarcation boundary and a natural fit for a PEP. Of course, your policy model must support defining microservices as targets, with attribute- and context-based access controls, in order for this to be effective.
DevOps

DevOps—which is a mash-up of the terms Development and Operations—represents a newer way of approaching application development, centered on collaboration between formerly siloed software development and operations teams. By using automated toolsets and rapid cycle times, this approach—which does require cultural and process changes—has been proven to help organizations dramatically increase their deployment velocity, release quality, and business value.

Ultimately, DevOps is about getting code quickly and continuously into production. Quite frequently, DevOps teams adopt Continuous Integration (CI) and Continuous Delivery (CD) approaches, which utilize a high degree of automation throughout the build, test, release, and deploy phases of DevOps. This automation is tied into the “infrastructure as code” approach, in which not only is the software application automatically built and deployed, but so is the virtual infrastructure upon which it’s running—and both of which are described by configuration (code) in a repository.

This may sound complex—and it is—but it’s given organizations the ability to get applications to market quickly, increases team productivity, stabilizes production environments, increases customer satisfaction, and provides consistent code deployments—ultimately delivering business value.

Figure 18-3 depicts the multiple phases of DevOps. This is commonly (and deliberately) portrayed using the “infinity” symbol, which represents the continuous and never-ending nature of DevOps. A natural question, of course, is where security fits into the DevOps model. The answer—the only correct answer—is “everywhere.”
In fact, there’s a term and a set of practices devoted to applying security throughout DevOps, termed *DevSecOps*. This approach ensures that multiple aspects of security are properly incorporated into the software design, development, deployment, and operations. This is important, because traditionally security was an afterthought of development, with detrimental results. In contrast, when security is designed in and thought through at the forefront, security frameworks can be effectively woven throughout the DevOps cycle.

Note that while in this section we’re looking at DevOps from a narrow Zero Trust perspective, there is a much larger part of application security that sits outside the scope of Zero Trust—such as static code analysis, functional security testing, fuzzing/input validation, and library vulnerability management.

**DevOps Phases**

Let’s now look at the DevOps phases, and see how Zero Trust applies to them.

**Plan and Code**

From a design perspective, this phase is where security teams should collaborate with and educate application developers on their Zero Trust architecture, capabilities, and policy model. Giving application designers this knowledge will help them decide where they can rely on the Zero Trust platform and where they need to take responsibility. For example, a high-value application won’t need to implement MFA, device posture checks, or geolocation restrictions if it can rely on the Zero Trust platform to do so.

And, application designers may be able to leverage the Zero Trust platform to obtain additional user context, such as validation of roles or permissions. These could be consumed and enforced within the application, essentially making the application a policy enforcement point.

**Build and Test**

As application code proceeds through the build and test phases, this is a natural place for the Zero Trust system to use automated policies that grant access only to the right set of people and tools based on workload attributes. For example, a testing workload could be automatically spun up, and only be able to access in-progress application instances that are properly tagged as being in *test* mode.
Release and Deploy

These last steps of the release process will result in the application being placed into production, within a Zero Trust environment with a full set of policy enforcement. That is, all access to application services is controlled by policies, which are only granted to authenticated and authorized subjects. Depending on the degree of automation, Zero Trust policies may even control access to the production environment, for example, based on approved change windows or a valid Service Desk ticket.

Operate and Monitor

For this phase, Zero Trust will help ensure the stability of the environment, and control any administrative or troubleshooting access to production applications. It’ll also provide identity-enriched logs, ensuring that all access is properly associated with authenticated identities.

Considerations

DevOps is an interesting and relevant use case for Zero Trust because there are so many ways to tie it to, and get value from, Zero Trust. Even basic integration gives security and application development teams the opportunity to balance and share access control approaches and policies. Breaking down this traditional silo helps “bake in” Zero Trust integration throughout the entire application lifecycle.

Designing an application component (or microservice) to consume and enforce PDP-defined policies can influence application security, and deepen the impact and value of Zero Trust in the enterprise. In essence, this can allow an application to become, in some ways, its own PEP (depending upon how much Zero Trust policy or context it can consume from the PDP). This can be woven throughout DevOps cycles—where the set of policies that are supplied to the application (and therefore enforced) will be altered to match its current phase.

Next, consider the use case we alluded to previously, where manual release and deployment of code may represent a security weakness. By applying Zero Trust policies throughout the release and deployment phases, organizations can ensure that this high-impact access is properly controlled, for example, by enforcing approved change windows.
Finally, managing access to your organization’s software designs and source code is a core Zero Trust use case. These assets are clearly valuable and like any high-value data deserve to be properly secured, with access controlled by a PEP.

**Recommendations**

The purpose of DevOps is to provide a high-velocity, high-quality, high-reliability means of delivering application code into production, in marked contrast to the traditional Software Development Lifecycle (SDLC). DevOps is better suited to many of today’s quickly changing environments, where getting incremental code into production quickly is what often drives business value.

Because Zero Trust systems are themselves inherently dynamic, and inherently responsive to user, service, and infrastructure context, they are a good fit for use within a DevOps environment. A Zero Trust system can be connected to an organization’s DevOps platforms, and automatically adjust access as workloads flow through the full application lifecycle. Zero Trust also helps improve on and automate security around areas that may still require manual steps, for example, by automating access controls based on approved change windows.

DevOps and Zero Trust are both modern and effective approaches, and organizations should definitely look at how they can be integrated together in support of each other.

**Mergers and Acquisitions**

From a security and technical perspective, Mergers and Acquisitions (M&A) represent complex and often lengthy projects which must attempt to reconcile two previously independent enterprises. These enterprises’ IT and security infrastructures were built and evolved completely separately, utilizing technologies and architectures in ways that may be incompatible (or at least difficult to reconcile). These two organizations will almost certainly have duplicate solutions in many areas, and will likely have overlapping network IP address ranges that are bound to cause problems—a too-common occurrence in our IP v4-centric world.

Recall that Zero Trust platforms, in addition to providing security, also provide a unifying or normalizing layer on top of heterogeneous resources and networks. This has many benefits within a single enterprise, as we’ve discussed throughout this book, and it also helps to quickly enable network access in an M&A scenario.
Specifically and tactically, a Zero Trust system can provide near-immediate IT access across domains, in order to quickly enable joint administration. Likewise, it can enable precise and secure user access to specific business-critical applications, for example, financial management systems. Given this value, let’s take a look at the next level of detail.

**Considerations**

If one of the two enterprises already has a Zero Trust deployment in place, an M&A activity should be an obvious catalyst to expand its usage, especially if it’s the acquiring company (which tends to be larger, and more able to impose its IT and security infrastructure). However, even if the acquired company is the one with Zero Trust, the merged enterprise can still use that platform to at least accelerate the integration activities. The value of this should be apparent—no other security or remote access solution can as rapidly, reliably, or precisely bring together two disparate (and often conflicting) enterprises.

A Zero Trust approach may also represent an opportunity to avoid significant costs and efforts that are typically needed to ultimately merge, normalize, or de-conflict the networks. For example, it may not be necessary to deploy a WAN to link the enterprise networks, if all users and servers get the access they need through a Zero Trust system. And, the enterprises may not need to de-conflict overlapping IP addresses on the networks if the Zero Trust system supports access mechanisms that can compensate for this.

As you approach this use case, think about which resources people need immediate access to, where they are located, and how they’re protected today. Of course, each enterprise will have its own identity provider, IT management, and security tools—all of which Zero Trust can help normalize nearly immediately.

**Recommendations**

If you have a Zero Trust solution in place and are acquiring a firm, using it to accelerate the transition should be a “no-brainer.” If you don’t have such a solution already deployed, but the company you’re acquiring does, strongly consider using that Zero Trust platform to help with your transition. At the very least, your employees will be able
to use it to access resources belonging to the acquired company. And, you should be able to extend that system easily to grant acquired company users access to your firm’s resources, for example, by deploying a PEP in your company’s network. Ideally, in this situation, you can use this to make the case for adopting Zero Trust within your larger enterprise—the acquired firm has demonstrated success with it, and you should be able to leverage this quickly to deliver value.

Finally, don’t forget about the server-to-server use case. In many cases, there are data synchronization or export/import activities that require production servers in one domain to communicate securely with production servers in another. Zero Trust systems make it possible to achieve this quickly and securely, without putting either organization at risk.

**Divestiture**

Divestiture, in which an enterprise spins out part of its business into a newly formed independent entity, typically represents a complex challenge for IT and security, but is also an exciting opportunity. The new company will certainly inherit part of the IT and security infrastructure, often including physical assets such as hardware, networking gear, networks, and buildings. While these assets are the definition of “brownfield” environments, IT and security teams will also typically be empowered to select new systems and tools to fill in gaps, or replace elements that must be decommissioned over time. This should give the IT and security teams the opportunity (and the budget) to deploy a Zero Trust system for this new environment.

In addition to deploying an infrastructure for a new company, there’s also another aspect of a divestiture that lends itself to Zero Trust—the transition period. In nearly every divestiture, the business and legal transaction occurs before much of the technical work can even begin. Even as the firms are legally separated, they’ll still be tied together by numerous technical systems, data flows, and business processes, which typically take months to unwind. Zero Trust can be used quite effectively to provide precise access control to critical resources that were “left behind” during this transitional period—keeping users and servers productive while preventing unauthorized network access. As the new company transitions off systems, one by one, access to them can be easily terminated via a simple policy change within the Zero Trust system.
Full Zero Trust Network/Network Transformation

This is a fitting use case with which to conclude the chapter, and to tee us up for the discussion about the journey of deploying Zero Trust, in Chapter 19. This scenario is in some ways a composite of the scenarios we’ve just covered, and in some ways is considerably different from all of them.

The most important difference is that going “full Zero Trust” involves a shift in networking philosophy, that of taking all your users “off net” and requiring use of the Zero Trust system to access any enterprise resource. Interestingly, the abrupt COVID-19-driven shift to a predominantly work-from-home user population in early 2020 accelerated many organizations’ readiness to make this change. The biggest mind shift associated with this is the realization that the problem to be solved isn’t “remote access”—it’s just “access.” In fact, taking a unified approach to securing all access is what underpins much of the value of a Zero Trust environment.

The term “full Zero Trust network” implies a large and comprehensive scope, but in practice, you define the limits and boundaries for your initiative—not every Zero Trust journey has to end with microsegmentation for every single resource. In some ways, it’s advantageous to use the more ambiguous term “network transformation,” rather than the term “full Zero Trust,” which may lead some people to an incorrect conclusion.

So, as you go through this process, be sure to define limits and have a realistic vision for your end state in mind. In our experience, we’ve most commonly seen enterprises envision their Zero Trust end state to be as follows:

- All users are off the enterprise network.
- Most private services are protected by PEPs, typically using the enclave-based model.
- Some SaaS services may be protected by PEPs.
- There may be some sets of services using microsegmentation.
- There will be some implicit trust zones in which services are running.

The implications of this are, of course, the changes and benefits that we’ve been advocating throughout this book. The elimination of the trusted enterprise network gives the organization much more resiliency, and reduces both the attack surface and the blast radius. Users have “always-on” Zero Trust access, with dynamic and context-sensitive policies evaluated to provide them with sufficient access to be productive,
while enforcing the principle of least privilege. This principle ensures that all access is explicitly granted by policies, increasing the organization’s visibility of network and computing assets. And, the enterprise IT and security infrastructure is integrated at a data and process level, increasing efficiency and effectiveness. Let’s take a look once again at the conceptual Zero Trust architecture diagram we introduced early in the book, in Chapter 3, shown again in Figure 18-4.

Figure 18-4. Zero Trust Architecture

This diagram shows the ways in which the representative enterprise from Chapter 3 has chosen to deploy their “full Zero Trust” architecture. They have incorporated most of the approaches discussed throughout the book, addressing their concerns and obtaining their desired benefits. Let’s examine how they approached this.

Their PDP is connected with their enterprise identity provider (IAM), of course—this is a foundational prerequisite. And their PDP is also integrated with other IT and security infrastructure elements, such as their MFA, SIEM, GRC, endpoint management, and PKI systems. There are a set of distributed PEPs throughout their infrastructure—many
of which are enforcing access into resource enclaves. The organization is also using local user agent PEPs on most users’ devices, and has deployed PEPs directly onto some servers as well. Note that there’s an encrypted PEP-to-PEP connection between the PEP in the DMZ and the PEP in front of the implicit trust zone—this is a configuration supported by some Zero Trust platforms.

Their Zero Trust system secures access to both SaaS and IaaS resources, and the PEPs in their IaaS environment use dynamic attributes (metadata) on the workloads to make access control decisions. Note that in their branch offices, they’ve deployed the PEP in a way that manages resources and users from an IoT-style perspective. That is, devices (and users) on that network can access (and be accessed by) Zero Trust-protected resources.

Finally, note that not all elements on the network are in scope for the Zero Trust solution. For example, there are implicit trust zones (resource enclaves) in the IaaS environment as well as between the resources in the enterprise network. Also, note that while admin access to the web server in the DMZ is controlled by a PEP, customer access to other services on that server is outside the scope of the Zero Trust solution.

**Considerations**

Clearly, full Zero Trust is a big initiative and even with top-down endorsement and support is going to be a technical and organizational challenge. In fact, not all enterprises will be ready for it, especially as a first Zero Trust motion. We’ll be exploring this aspect further in Chapter 19, but before we do that, we want to make some recommendations.

**Recommendations**

While a large-scale network transformation project may not be possible initially, we do want to reinforce that reducing users’ network privileges is an important goal; in fact, it’s one of the most important things you can do as part of your Zero Trust initiative. It can be achieved incrementally, so even if you accomplish this subnet by subnet (or VPC by VPC, or application by application), it provides value.

We acknowledge that enterprise networks are complex and that there are many in-place elements that may appear to act as constraints or barriers. But this doesn’t necessarily have to be the case. For example, consider an office with printers that users
get implicit access to when they’re on premises. This access can easily be provided by a Zero Trust policy, and this requirement shouldn’t be an impediment to adopting Zero Trust. In fact, in some cases, in-place components can be leveraged to enable Zero Trust. One of our enterprise clients had a NAC solution that was already deployed across 50+ branch offices. As they rolled out their Zero Trust agent to users’ devices, group by group, they configured the NAC to assign users in the relevant groups to the guest VLAN instead of the employee VLAN, effectively taking them off net. The beauty of this change is that end users didn’t even notice—they remained fully productive and able to access all their applications.

In some ways, each of the previous six use cases is a microcosm of the ideas, approaches, and challenges of the full Zero Trust network scenario. This is what makes this such an interesting set of problems, and is also yet another good reason to begin with a more focused use case rather than full-bore Zero Trust. By starting with a smaller scenario and user population, you don’t have to strategically solve every problem “at scale,” and yet you’ll be learning and creating things (policies, teams, processes, etc.) along the way, which will make it much easier for you to achieve this larger use case over time.

**Summary**

To recap, in this chapter, we analyzed seven different scenarios for applying Zero Trust in the enterprise. We’ve mentioned most of these use cases throughout the book, but this chapter gave us the opportunity to examine each of them in depth and to do so with the benefit of building on the knowledge and context we’ve learned in the preceding 17 chapters. As we surface from the details of these use cases, take a breath and a step back—in Chapter 19, we’ll be looking at how your organization should approach Zero Trust from a program and initiative perspective, in order to ensure success.