SANS

Analyst Program

Whitepaper

Zero Trust: Charting a Path to Stronger Security

Written by **Mark Ryland** and **Ashish Rajan**
Contributor: **Quint Van Deman**

August 2023

aws

Introduction

Security has become a top priority for organizations looking to build customer trust, enhance workforce mobility, and unlock digital business opportunities. However, the traditional approach of defined security perimeters that separate "trusted" from "untrusted" network zones has proven to be inadequate. Today's distributed enterprise requires a new approach to ensuring the right levels of security and accessibility for systems and data. Increasingly, zero trust is being described as the solution.

Gartner, a company that delivers actionable, objective insight to executives and their teams, predicts that by 2025, over 60% of organizations will embrace zero trust as a starting place for security.[1]

Zero trust is a journey that's different for every organization. For some, the journey is a natural evolution of cybersecurity in general, and defense in depth in particular. For others, it's driven by policy considerations, and by the growing patchwork of data protection and privacy regulations across the globe.

Regardless of the rationale—and despite the hype that surrounds the term—zero trust can meaningfully improve both technical and business outcomes. However, implementing a zero trust architecture is a process that requires careful consideration. Organizations often find themselves asking, "What exactly is zero trust?," "How do I get started?," "How do I make continued progress?," and "How do I demonstrate return on investment (ROI)?"

This chapter explores these important questions and cuts through the zero trust hype with best practices for designing a successful strategy that supports secure access to resources with a broad range of evaluation factors.

Defining Zero Trust

While zero trust has quickly grown from concept to strategic priority, there may still be some confusion around exactly what it is. Definitions vary, but zero trust is essentially a security model and associated set of mechanisms that focus on providing security controls around digital assets that don't solely or fundamentally depend on traditional network controls or network perimeters. Zero trust encourages you to incorporate a wide range of context about any particular access request, including identity, device, data, behavior, and more, so your systems can make increasingly granular, continuous, and adaptive policy-based access control decisions (see Figure 1).

[1] *Gartner, "Gartner Predicts 2023: Zero Trust Moves Past Marketing Hype Into Reality,"* John Watts, Jeremy D'Hoinne, Dale Koeppen, Charlie Winckless, 6 December 2022. *GARTNER is a registered trademark and service mark of Gartner, Inc. and/or its affiliates in the US and internationally and is used herein with permission. All rights reserved.*

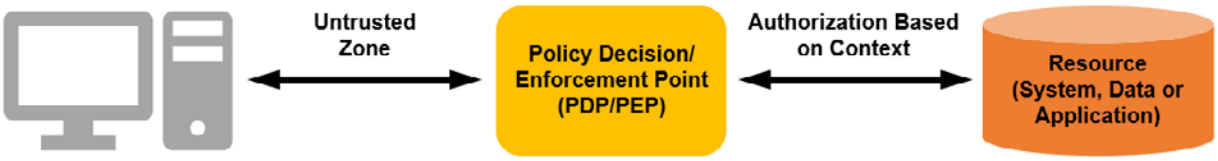

Figure 1. Zero Trust Access

The focus on access control is important because, although authentication and associated concepts like identity federation have been reasonably modernized and centralized, authorization typically remains spread across countless downstream systems. Authorization rules exist in access control lists, table grants, in-app permissions, and other similar constructs in ways that are difficult to configure and manage, much less consistently track and audit. When you distill zero trust down to its essence, ubiquitous and increasingly centralized authorization is one of the fundamental problems it aims to solve.

In practice, zero trust can also be thought of as the convergence of networking, identity, and security. Ideally, in a zero trust architecture, networking and identity-based controls aren't just simultaneously present and configured, they're actually aware of one another. An illustrative example of this is an Amazon Virtual Private Cloud (VPC) endpoint. VPC endpoints provide private network connectivity to AWS services from your own virtual private cloud, and allow you to specify access control policies. These policies and their associated enforcement engine understand not only the network, but also the identities and resources that are flowing across this border network control. They can make authorization decisions that consider this converged context.

Additionally, zero trust allows previously siloed security capabilities—such as the management of unified endpoints, vulnerabilities, service ownership, identity, and everything in between—to share data, signals, and telemetry to make more informed decisions. Improvements can come from both declarative policies that consider cross-silo factors, and from machine learning-powered processes that identify anomalous patterns or behaviors and either suggest policy enhancements to administrators or dynamically adjust authorization decisions based on risk. Convergence in these areas will take time, but it will serve as your North Star on the journey to zero trust.

> **"Zero trust itself isn't the goal; it's the how, not the what." —Mark Ryland, Director, Office of the CISO, AWS**

Tightening your focus from "security for security's sake" to objectives such as end user mobility, digital transformation, and customer trust—and the technical use cases that empower them—can help you move beyond going through the "we need to do something about zero trust" motions and articulate the need to invest time and resources in ways that relate to the business. This is important, because it makes it possible for you to stay focused on the fact that zero trust is all about facilitating desired business and technical outcomes.

Foundations and Fallacies

Zero trust requires foundational security capabilities to be in place. However, existing guidance often suggests a level of comprehensiveness, even perfection, across these foundations that can make even getting started feel like a Herculean task. It's important to understand which foundational capabilities are truly critical on your journey to zero trust and to avoid common fallacies along the way.

Foundation #1: A Solid Approach to Identity and Access Management

Identity is arguably the most important contextual factor in a zero trust authorization decision. Whether the primary actor is a user, an application, or a device—and whether the resource being accessed is on premises or in the cloud—prioritizing the deployment of several specific identity and access management (IAM) capabilities is key. These include:

- **Multi-factor authentication—**Modern multi-factor authentication (MFA) solutions, such as FIDO2 hardware-based security keys and associated processes for distribution, enrollment, and ongoing management, are vital to your zero trust efforts. The use of FIDO2 security keys, in particular, not only provides a high level of authentication assurance for zero trust authorization decisions, but also offers benefits such as phishing resistance. It also strikes an excellent balance between security (e.g., private keys that can never leave the device), usability (e.g., the user simply taps the device to authenticate), and interoperability (e.g., support that's automatically baked into modern operating systems and browsers via the WebAuthN web standard).
- **Single sign-on (SSO)—**Your MFA implementation should be paired with the services of an SSO/federated identity provider. Support for modern identity protocols that includes OpenID Connect (OIDC) for authentication, and System for Cross-domain Identity Management (SCIM) for replication of identity-related information is essential. This support is typically provided by most top-tier IAM solutions, and you can prioritize support for Security Assertion Markup Language (SAML), Kerberos, and other older protocols according to legacy and migration needs.
- **Identity governance processes—**Verify that your IAM capabilities include well-functioning embedded or surrounding processes for identity governance (e.g., covering joiners, movers, and leavers in enterprise group management). These processes, and the identity groups and attributes they control, are not only vital to authorization decisions but also serve as the basis of resource ownership information (as you'll see in Foundation #3 below).

Foundation #2: Unified Endpoint Management (UEM)

Understanding the health and security posture of a user's device is typically the second most important contextual component in a zero trust authorization decision.

You need to be confident that an endpoint is in a proper state before allowing it access to corporate data and resources. UEM solutions support this confidence by providing capabilities that include device provisioning, ongoing configuration and patch management, security baselining and telemetry reporting, and device cleansing and retirement.

Focus on the form factors most relevant to your workforce. This typically means starting with corporate-issued laptops and desktops, followed by mobile devices and cloud desktops. Depending on your organization's business needs and constraints, you may wish to consider allowing access to less confidential systems and data from uncontrolled endpoints as a risk-based decision (e.g., if your organization has a bring-your-own-device policy). Access to sensitive data from uncontrolled systems should be avoided wherever practical, in the absence of compensating controls such as the use of a virtual desktop solution or a secure enterprise browser.

Foundation #3: Resource Ownership Tools and Processes

Successful zero trust implementation requires a reliable system for cataloging the enterprise resources being accessed, and understanding who owns them. In this context, the "who" may not be a single individual but may instead be represented by a flexible grouping mechanism such as a ticket queue. Properly managed ticket queues have owners (who can change seamlessly over time), natural workflows, escalations, priority definitions, and other mechanics that help keep resource information accurate and can flexibly adapt to reorganization or reassignment as ownership of a given resource evolves. If your organization doesn't have extensive rigor around ticketing, you can use alternate mechanisms, such as email distribution lists. However, it is important to keep the maxim "When everybody owns it, nobody owns it" in mind when employing one-to-many mechanisms.

Your source of truth around ownership needs to provide, or be closely integrated with, workflows that facilitate access requests, associated approval decisions, and regular human reviews by responsible parties (i.e., "baselining"). Although some types of access can be inferred from attributes, job roles, and group memberships, ad hoc requests often outnumber rule-based access grants by a wide margin. These workflows should support an individual (or a proxy) requesting access to a given resource, which is then routed for approval, memorialized with descriptive data about ***why*** the access was needed and approved, and regularly revisited to verify that the need still exists. In time, this source of truth will contain the bulk of the information needed within the organization to answer the question, "Who can access what?" which will be used for both authorization policies and audit/compliance.

In addition to a technical repository, your organization should agree on an appropriate governance model for this kind of critical data that provides answers to questions about who can access what: Resource owners? A central team? A combination of the two? The answers don't need to be uniform across the entire organization, but your governance model should be clear and uncomplicated.

Foundation #4: Data Classification

Identifying, protecting, and managing access to your organization's core asset—your data—is an important step on the path to zero trust. However, not all data is created equal. You need visibility into the data you're collecting and storing in order to determine the right levels of data importance and sensitivity. Investing in data classification can help you divide information into predefined groups that share a common risk, and identify the corresponding security controls required to secure each group.

Access to data based on classification will help you prioritize incremental efforts to implement zero trust capabilities. Once zero trust improvements have begun, data classification can also help limit the potential exposure of data to a limited set of users and make security events that require further investigation more straightforward to manage. Encryption of data at rest and in transit adds another layer of security to classified data when it's being stored or is required by a user.

Although data classification is relatively simple to apply technically, it's important to set the right expectations and approach. Focus on iterative efforts geared toward constant progress, rather than waiting for anything like perfection. Full data classification can be an expensive and cumbersome activity for organizations that have been storing data for a long time (e.g., since before digitization). As you begin to apply the zero trust model and data classification to your organization's environment, you may decide to simplify the task by setting a time limit (such as two or three years), before which all otherwise-unclassified data is categorized at the least sensitive but nonpublic level. That makes the job simpler and more realistic (without a major impact on risk) because important data types, such as personally identifiable information (PII) or sensitive intellectual property, may already be classified.

Foundation #5: An Established Security Data Lake or Unified Logging

Zero trust architectures and technologies provide additional trust signals that result in more valuable data in security logs. However, this additional data needs to be centralized and standardized to realize its full benefits. Normalizing security telemetry across various security products and services is a key step toward the converged operation of previously siloed security capabilities. Instead of dealing with a variety of proprietary formats, the unified storage and formatting of data simplifies findings enrichment and incident response activities almost immediately and can quickly evolve into a powerful source of insight and continued progress in reducing access privileges.

The Open Cybersecurity Schema Framework (OCSF)[2] is an open standard designed specifically for this purpose. It provides a common language for the kind of security telemetry typically used in threat detection and investigation and has the broad support of well-established security technology providers. Licensed under the Apache License

[2] "Understanding the Open Cybersecurity Schema Framework," Github, May 2023. https://github.com/ocsf/ocsf-docs/blob/main/Understanding%20OCSF.pdf

2.0, OCSF is agnostic in storage format and data collection and can help you minimize the amount of extract, transform, and load (ETL) processing required during ingestion.

Before you can start sending all your organization's security and adjacent telemetry data to a common repository, that repository needs to be properly established. Start by picking a standard storage pattern for the data (preferably based on OCSF or a similar framework) and a raw storage repository—such as Amazon Security Lake, which natively supports OCSF—that can scale to meet current and future capacity and analytical performance needs, based on projected growth.

Be deliberate about your storage hierarchy pattern, and store data consistently. If one tool stores data in a region/host/date hierarchy, but another chooses date/region/host, the queries necessary to join these data sets may be unnecessarily difficult. Finally—although it's important for this core capability to exist—you don't need to wait for all the log sources across your organization to be fully integrated. Instead, these sources can and should be enumerated, prioritized, and integrated opportunistically, with care taken to demonstrate overall system intelligence improvement with each integration.

Foundation #6: Incident Response (IR) Testing

Once you've achieved a reasonable level of zero trust maturity, you can expect to prevent more security events and increase your threat detection capabilities due to an increase in the quantity and quality of security-related signals coming into your security tooling. However, an effective and enhanced IR process that takes advantage of these new data sources is important so you can identify and remediate even minor security events quickly (see Figure 2). This will allow you to disrupt the sequence of events that can escalate an initial incursion into a more high-impact incident.

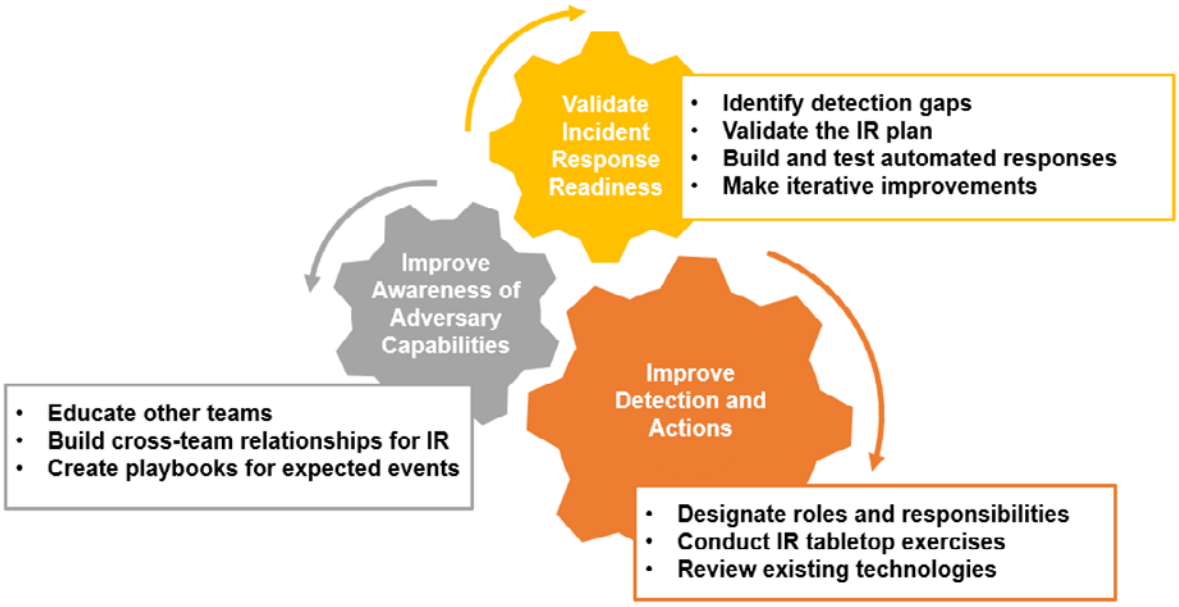

Figure 2. Strengthening Incident Response Readiness

Regardless of the IR framework or methodology your organization aligns with, you should test your IR plan regularly. Tabletop exercises, simulations, and red teaming provide opportunities to practice IR in realistic settings, uncover tooling and capability gaps, and build the experience and confidence of incident responders.

Fallacy #1: You Can't Start Without a Perfect Inventory of Systems, Identities, and Data

When it comes to making progress on zero trust, a ***perfect*** source of truth about your environment may be ideal but is not realistic. Accurate inventories have eluded traditional on-premises environments for decades. Configuration management databases (CMDBs) typically have poor data hygiene. Additionally, discovery tools are often cumbersome to deploy, and organizations struggle to use them to comprehensively capture assets due to existing network controls and segmentation. So, although you need to establish a source of truth, you can divide the effort into a scope that makes a "good enough" inventory quickly achievable, so you don't make the all-too-common mistake of allowing "perfect" to be the enemy of "good." Organizations that are all-in on the cloud, or have heavily migrated to it, may not find achieving inventory accuracy as daunting. Cloud environments significantly ease the process via descriptive application programming interfaces (APIs) and inventory services that allow you to instantly query the running state of your environment via the control plane in a way that's more accurate and up to date.

Fallacy #2: You Can Buy a Product and Quickly "Check the Box" on Zero Trust

Zero trust is a security model, not a product. Although you almost certainly will consume products and services from one or more vendors, you shouldn't lull yourself into thinking the journey to zero trust is as simple as buying and deploying a product that claims to solve your problems. Losing sight of this will, at best, lead to additional expense that doesn't fundamentally change your security model and isn't tied to business outcomes. At worst, approaching zero trust in this way can distract you from your true objectives and provide a false sense of being "done" when, in reality, little to no security improvement has been made.

Fallacy #3: You Need a Clear End-State Vision from the Start

Developing a general North Star vision is important, but don't expect a perfectly clear view of your journey's end before it begins. Careful evaluation of what works for your organization—and what doesn't— along the way precedes the ability to definitively outline your end state. Adjustments will undoubtedly be needed as you make progress and gain insight. Take a flexible approach to initial architectural diagrams and technical standards that depict what "good" looks like and be ready to adapt them as your efforts solidify and you become better-informed. Setting your focus on immediate needs and considering how you can make incremental security improvements that allow for value recognition, real-world experience building, and continuous progress toward an authentic zero trust future will keep your efforts practical, and help you avoid getting hung up on hypotheticals.

Fallacy #4: You Will No Longer Need a Traditional Network Perimeter

You should think of zero trust as largely additive to existing security controls. Network controls are well-understood, broadly deployed, and generally help demarcate an organization's enterprise resources. Network location is also among the important pieces of context that can be evaluated during authorization decisions in a zero trust architecture. And although these decisions must be evaluated and enforced from the edge to deep within the core, traditional network perimeters are some of the first and most logical enforcement points your organization can choose to enhance to take advantage of zero trust access control, because they already exist at various points throughout the network. Recognizing the idea that traditional network controls aren't relevant as a fallacy can help you avoid unnecessary power struggles that may arise when one function or department feels they may be obviated by, rather than integral to, zero trust efforts.

Common Use Cases

There are a number of common use cases that can benefit from the enhanced security provided by zero trust. It's important to work backward from the specific use cases that apply to your organization to determine the optimal zero trust patterns, tools, and strategies that can help you achieve meaningful security advancements. Approaching each use case with an eye to the big picture facilitates progress.

Use Case #1: Human-to-Application

Many organizations start with the human-to-application use case. It's commonly referred to as Zero Trust Network Access (ZTNA) and is often confused with zero trust in its entirety. Although preceding sections of this chapter have largely related to this primary use case, many aspects of the foundations and fallacies apply equally to the additional use cases described below.

In the human-to-application use case, zero trust principles are used to allow employees to access the internal applications they need to do their jobs from anywhere, without relying on a virtual private network (VPN). Although this use case is most often focused on workforce mobility and productivity, it can help your organization realize additional benefits, such as a relatively effortless transition from dated application-level identity protocols, such as Kerberos, to modern identity standards, such as OIDC.

Use Case #2: Service-to-Service

The service-to-service—or machine-to-machine—use case helps you consider pathways within and between workloads, and minimize those that are unnecessary (particularly those that lead to data). Although the human-to-application use case controls how a given actor reaches an application, this use case often controls the resulting flows within an application or between microservices that are composed into an application.

It can be useful to separate efforts related to custom-built services from those focused on services consumed through your cloud provider—at least until you can determine whether they will be implemented with the same or disparate technology and associated controls.

Use Case #3: Internet of Things and Operational Technology

This increasingly common use case supports organizations that are pursuing the interconnection of devices, machines, facilities, infrastructure, and processes outside the traditional network perimeter as part of digital transformation. Internet of things (IoT) and operational technology (OT), also known as Industrial internet of things (IIOT), devices often transmit telemetry and predictive maintenance information directly to the cloud, requiring the application of security controls that extend beyond the traditional perimeter approach to protect workloads.

Use Case #4: Operator-to-Infrastructure

Many organizations are interested in moving beyond development and operations (DevOps) to a fully automated IT environment that requires no hands-on operations work distinct from software development and automated pipelines for testing and promoting code to production (NoOps). However, although NoOps can help you achieve a faster deployment process, it is a journey in and of itself. Regular or break-glass style operator access—which often involves privileged levels of access to operating systems (OSs), database engines, or container infrastructure—needs to be supported along the way, and likely forever to some limited extent. This makes the enhanced access controls afforded by zero trust an imperative. This use case is best approached separately from end-user access, due to divergent tools and access patterns. For example, a user accessing a system through a web application has different security implications compared to direct access to an interactive shell through a protocol such as Secure Shell (SSH).

Use Case #5: Human-to-Data

Organizations of all sizes are using data to enhance customer experience and build new revenue streams with artificial intelligence (AI), machine learning (ML), and advanced analytics. Many of these advancements are driven by data scientists whose work requires access to large amounts of raw data, much of it highly sensitive. Today's binary approach to access runs counter to the zero trust model. Thinking of the difficulty involved in "keeping humans away from the data" helps highlight the need for more granular and flexible preventive and detective controls in this area.

Use Case #6: Authorization Inside Custom Applications

Zero trust involves making access control decisions on individual data elements, artifacts, and other small resources that number in the millions or billions. Although patterns vary, these small resources—think single rows or even cells in a database—are often conceptually modeled at a lower level within custom application business logic

that is more granular than the cloud services or data repositories that store them. For example, a single Parquet file containing records in JavaScript Object Notation (JSON) format in an object storage service might contain thousands or even millions of records, each requiring unique permissions. Most organizations will begin approaching zero trust at higher levels that involve coarser authorization decisions. However, it's important to keep the most granular use cases in mind and verify that your organization's zero trust tooling is capable of further development to cover more granular access controls in the future.

A Key Consideration

Early in your zero trust journey, you'll likely come to a fork in the road as you consider a question that's basic to your overall strategy: Do you want to achieve consistency of ***outcomes*** or consistency of ***implementation***?

A consistency of outcomes strategy views zero trust as a model and a set of ideals that should be implemented with all of the features available in each major compute environment used. Organizations taking this approach are willing to accept some level of heterogeneity in tooling, templating, and reporting to achieve desired security outcomes. These outcomes include things like development and operational efficiencies, integrations, and inherent capabilities or other benefits that would have to be sacrificed or duplicated when a consistency of implementation approach is used.

A consistency of implementation strategy prioritizes standardization and the efficiency it provides the entire organization, over an optimal quality of implementation for each narrower domain. This typically requires ignoring native or default capabilities in favor of solutions that attempt to address the overall requirements of the organization. This approach has some advantages. However, it can lead not only to less tailored and optimized results in a given domain, but also to the duplication of features that can leave some teams confused about the tooling choices, as they are unaware of the broader context and the expected value of organization-wide standardization.

Trade-offs are familiar to most organizations and technology leaders. One example: Complex, heterogeneous environments (such as those running on both Windows and Linux) can either be managed by distinct teams with distinct skills, tools, and modes of work, or those environments can be managed by a uniform abstraction that operates under the premise that "patching is patching," regardless of the OS. Neither point of view is incorrect, but this decision should not be made lightly, as it may not be easy to reverse down the line. When choosing your approach, be careful to avoid common estimation errors. Examples include undervaluing the inherent capabilities provided by cloud environments, overvaluing the flexibility and abstraction provided by a consistency of implementation approach, and underestimating the time and skills necessary to define, build, and maintain zero trust for more than one environment.

Getting Started

Organizations can quickly become overwhelmed by the scope of their zero trust journey. Working to establish the foundations described above, while avoiding mistakes that can result from common fallacies will support your efforts as you make small, well-defined steps toward zero trust. Several best practices can help you chart a path to success:

- **Articulate goals—**Clearly define why you're moving toward zero trust and communicate the goals your organization aims to achieve. This will be more

valuable than describing a technical architecture meant to represent a future state. List key stakeholders (e.g., business users, developers, C-level leadership, board of directors, and security administrators) in your organization, and write a concise summary for each one that articulates why they should care about your zero trust efforts and how those efforts will directly benefit them. Be prepared to consistently deliver, reinforce, and refine these messages as your journey to zero trust progresses.

- **Work on use cases—**Although there are numerous use cases—as detailed above—most organizations should start on "the big two" use cases: human-to-application and service-to-service. These use cases are typically the easiest to separate into a manageable amount of work, they naturally fit back-to-back, and they're straightforward in terms of visibly measuring value and progress. They also tend to involve different groups within the organization, allowing progress to be made in parallel.

  Human-to-application (or ZTNA) is typically expressed as something like, "allowing workforce users to access internal applications from any coffee shop in the world, no VPN required." This use case forces the organization toward the recognition that strongly authenticating a human, evaluating the health and posture of their device, and continuously assessing security state as part of each access request are now the most critical parts of an authorization decision. It is important to focus on this use case early because it directly touches and improves the experience of everyone in the organization who will use it to get their work done every day. One major benefit of starting with the human-to-application use case is that the business leaders who are prioritizing and funding the effort will have a very real and tangible appreciation for the transformation, since they too are users of the new capabilities.

  The service-to-service use case (or machine-to-machine) involves tackling the relative lack of east-west network controls and visibility that often plagues traditional networks and their associated perimeter-based security models. By being deliberate about which components you expect to talk to which other components and how, your organization can disrupt the lateral movement that's often a key part of a security event, while also making the detection and remediation of any network intrusion, however minor, much simpler. By doing so, you can realize a very real and measurable risk reduction.

  The service-to-service use case will also clarify the decision between consistency of outcomes and consistency of implementation, given the stark difference between traditional on-premises networks and API-driven cloud connectivity patterns and the fact that service identities are generally a "solved problem" in the cloud, while root of trust and secrets management and distribution challenges are still meaningful obstacles on premises. Organizations that are willing to move toward the consistency of outcomes approach will likely find that the service-to-service patterns available in the cloud make it possible to completely rethink

traditional implementation patterns and reduce the surface area of compute services, while dramatically simplifying the experience for developers, network engineers, auditors, and security professionals alike.

- **Develop living reference architectures—**Develop an initial, dynamic architecture depicting what "good" looks like for each use case. This will allow you to begin building, yet be ready to adapt as your efforts progress. These reference architectures should be thought of as living artifacts that will continue to evolve. Beyond acknowledging that things will change, this will encourage teams to think about templatizing the architectures for consumption over time.
- **Scope and build authenticity—**Focus your attention within use cases on making progress and gaining momentum. Start with a reasonably sized group of applications, where the business value of the data or the greatly increased convenience for users—or both—is worth the effort required to implement zero trust. By initially focusing on a small and meaningful set, you can refine the necessary technical and operational processes in a flexible and iterative way, while building the authenticity and experience necessary to expand efforts to an increasing percentage of your organization's IT environment. The department leading your zero trust initiative may wish to move one of their own applications or application groupings first to give others confidence that the team has already walked in the footsteps they're asking the rest of the organization to follow.
- **Consider retrofitting versus modernization—**Consider the relative effort and value of retrofitting zero trust into a particular application for a particular use case as-is-where-is versus building zero trust into the application as part of a broader modernization or cloud migration initiative. Although you should be careful about intertwining efforts such as zero trust, application modernization, and cloud migrations if they're already underway or planned, there may be an opportunity to implement zero trust with little to no additional effort.
- **Fuel the adoption with champions—**Think explicitly about rollout, adoption, and value creation as you start your journey. This is not a "build it and they will come" endeavor. Fortunately, there are natural incentives that will drive the rollout. Zero trust makes life easier for end users, so they will become your biggest advocates for getting applications onboarded. It makes life easier for developers by offloading security concerns that previously had to be dealt with in their application logic (or perhaps weren't being dealt with at all), and often providing a "free upgrade" to modern application identity. It produces real outcomes for security teams by increasing levels of assurance for application access and ultimately providing a pathway to shrink an abundance of network connectivity and surface area out of dynamic environments such as office buildings. When it's a win-win for everyone involved, the rollout will typically progress quickly, without the need for large-scale campaigns or program management of a forced "security mandate."

Driving zero trust adoption will take time and effort, as you begin to experience the implementation and its benefits. The team championing zero trust within your organization should be deliberate about partnering with the other stakeholders necessary to complete the initial waves previously described. However, once started, the steady growth of adoption should build momentum on its own, as users demand an improved experience across more enterprise assets, and engineering teams recognize the operational benefits of the implementation.

Measuring Progress

As with any strategic initiative, measuring progress, return on investment, and solution efficacy are key to quantifying the positive impact, maintaining executive buy-in, and justifying budget allocation and investment. However, the impact of zero trust often amounts to measuring what *didn't happen*—or what would *otherwise have happened*—if protections were not in place. Although it's impossible to measure these outcomes with perfect accuracy, you can present metrics that reasonably approximate these impacts. When combined with anecdotes and day-to-day hands-on experiences, these metrics can present a sufficient view of impact and progress.

A basic accounting of rollout progress provides a good starting point. Example metrics might include:

- The number of workforce users properly equipped to access zero trust-ready workloads and those that have the necessary MFA and/or managed devices
- The number of zero trust-enabled workloads, with breakouts for critical or highly sensitive workloads
- The number of security systems sending telemetry to the security data lake or other unified logging sink

For each metric, when the total number is known or reasonably approximated, each scalar value should also be expressed as a percentage, even if the denominator changes over time.

Next, you can strive to account for bad outcomes that were either prevented or minimized by additional zero trust controls. Metrics of this nature will typically require some level of additional labeling, computation, or analysis. Examples include:

- The number of security events prevented by zero trust controls that would not otherwise have been prevented (e.g., denies based on zero trust-specific context)
- The mean time to detect (MTTD) security events—for events that aren't prevented (zero trust should lower MTTD)
- The number of detected security events that were remediated before reaching sensitive data or systems (by lowering MTTD, we also should reduce—with the goal of zero—the number of significant security events)

- Rate of false positives within detected security events (by using a cross-cutting set of telemetry to make security detections, the false-positive rate should decline over time from the pre-zero trust baseline as the system learns)

If your organization has a calculated or industry-approximated per-occurrence dollar figure you are comfortable with, these metrics also can be expressed in terms of "estimated savings," with appropriate caveats. Any such calculation should attempt to account for both direct costs (e.g., external incident response avoided) and indirect costs (e.g., brand reputation or privacy-related fines).

Conclusion

The changing workforce landscape, shifting regulatory requirements, and a need for more precise and least-privileged access controls have led to zero trust becoming a pragmatic choice for IT security strategies. But the journey to zero trust is an iterative process, and it's different for every organization. By considering your own environment, establishing the right foundations, and avoiding common fallacies along the way, you can move beyond traditional security approaches and make continuous progress toward achieving strong levels of security for systems and data.

Sponsor

SANS would like to thank this paper's sponsor: