Bastion zero

FROM ZERO TO TRUSTLESS

A No-Bull Guide to Zero Trust Access

Introduction

The digital economy is built on credentials and powered by privilege. When a developer pushes code remotely, accesses a machine, or authenticates into a mission-critical system, what occurs is not just an exchange of data—it's **a transaction in the ultimate currency: trust.**

Each of these seemingly simple, everyday actions may feel routine, but the trust exercised is the lifeblood of an organization.

As organizations evolve their access strategies, they are constantly making **high-stakes decisions** that have to do with trust. Remote access technologies are one of the most sensitive and valuable assets used by many organizations. Whether their resources are located in the cloud or on-prem, security teams need to balance risk with employees' need to access them anywhere and anytime. **In a never-ending fight against breaches, how can organizations make the best decisions about access?**

This guide rethinks trust by examining the role of privilege in traditional remote access tooling, challenges to these approaches, and the growing need for trustless access. We'll break down the two phases of an attack—the user and system sides of privilege—and how trustlessness impacts access from the user's perspective.

"Zero Trust" Is a Misnomer

Almost all breaches can be attributed to a very **common attack pattern.** First, a user's credentials are compromised. Then, once the attacker has obtained those credentials, they look to leverage this access to exploit weaknesses in the organization's systems and applications.

Zero trust is the solution architecture proposed to address this attack pattern. Let's look into the true definition of zero trust and why existing systems are failing to live up to their promises.

The concept of zero trust is most commonly associated with eliminating the credential compromise phase of a breach. But this association does not provide the full picture. Before we get too far along, let's define zero trust.

ZERO TRUST DEFINED

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

National Institute of Standards and Technology (NIST)
Special Publication 800-207

The term "zero trust" is confusing because it is inherently a misnomer. The National Institute of Standards and Technology (NIST) defines the basic tenets of zero trust as follows:

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

Zero trust can sometimes feel performative. Every night before you go to bed, chances are that you probably lock your front door. However, logically, you know that if someone really wanted to break in, they could. Still, locking the door gives us peace of mind.

Many steps taken in security feel like locking a door. There are things we do as security professionals because they feel appropriate. However, we know deep down that if a hacker is determined to get in, **it's only a matter of time.**

Looking at the seven tenets from NIST, you may have noticed that the word "zero" never comes up. Access isn't given to *zero* resources. Authorization isn't given *zero* percent of the time. **Zero trust doesn't mean zero things are trusted.** Unlike the name implies, a lot of trust must actually be exercised. This is where the breakdown starts happening between what zero trust promises in theory and delivers in practice.

The doors are locked, but that doesn't mean hackers are staying away.

Evolving Security for Today's Workforce

At one point, traditional approaches like zero trust were fit for purpose. Nowadays, however, they have been stretched beyond any use case that was conceivable at the time of their design. Basically, the industry's interpretation of trust and privileged access hasn't caught up with the risks associated with **rapidly modernizing use cases.**

It's important to note some of the most pressing challenges facing security professionals today:

The Shift to Remote Work and the Cloud

In the cloud and remote world, permissions have become more complex. Although these challenges predate the pandemic, the expedited move to remote work and the cloud has certainly **compounded the risks associated with access.**

Security Turnover and Burnout

Due to the cybersecurity skills gap, security professionals have many options for their next career step, and turnover is high. Security careers are stressful and unrelenting, often leading to burnout. With tight budgets and priorities spread thin, teams are forced to keep the lights on with limited backfills and resources. Turnover is never easy to deal with, but it's especially taxing when it comes to IT security roles. **Adversaries don't slow down when a team has open roles to fill.** Limited staff still need to fight the never-ending battle against breaches, and burnout is common in these exhausting circumstances.

Shift of Responsibilities toward Engineering

The lack of resources has forced IT to share accountability across the entire organization, meaning **security is no longer just an "IT problem."** Engineering often manages access, especially in situations where there is no centralized strategy for managing access. Engineering teams are one of the stakeholders that are increasingly taking on the burden of responsibility when it comes to security.

Organizationally, different groups can agree on the need to limit privilege. However, in practice, things like productivity, efficiency, and deadlines are often used to justify maintaining the status quo. Sometimes it isn't even done on purpose—for example, **privilege creep** is extremely common. Privilege creep happens when the security team gives someone access to a resource and then forgets about it, never taking that access away when the user is done. If this goes on long enough, everyone eventually gets access to everything.

Ideally, organizations could completely eliminate all access to mission-critical systems and instead use log data and other observability tooling to debug issues and respond to incidents offline. The reality is that most organizations aren't there yet; responding to incidents and production bugs still requires providing engineers with access to mission-critical systems. This puts even more **pressure on the security teams** to identify tooling to be able to audit, log, and control access, while maintaining the status quo of existing workflows.

Is it possible for these expectations to exist in tandem with IT's expectation for security above all else? And we haven't even mentioned the pressure IT gets from leadership and the board to keep persistent adversaries at bay.

A 2022 Gartner survey found only 29% of IT professionals intend to stay with their current employers. This is over 10% lower than non-IT employees and is the lowest out of all corporate roles.

According to the 2022 Verizon Data Breach Investigations Report, "50 percent of breaches involved the use of either remote access or web applications." If organizations want to survive in a decentralized world, they must evolve.

Credential Theft and Single Points of Compromise

Credential theft occurs when the security of a system being used to authenticate users is compromised. In most cases, this authentication system is usually a **single sign-on (SSO)** provider, a certificate authority, or both. However, in some cases, they are "loose" credentials like SSH keys or database passwords.

Organizations must be careful about how they solve the problem of credential theft because solutions often inadvertently create a **single point of compromise.**

SINGLE POINTS OF COMPROMISE DEFINED

"Key enterprise central services that could be misused by an intruder or an insider to compromise critical portions of an enterprise's computing environment. When determining what services should be classified as a single point of compromise, consider services where a compromise would allow login or root login on many assets within the environment,ensuring a complete ownership of the institution's environment by an adversary."

SANS Institute

Compromising an SSO provider or certificate authority allows bad actors to issue credentials to themselves. Once this happens, bad actors can log into any and all systems, making it easy to move laterally through the environment undetected. Once in a system, attackers may escalate privileges until they own the entire architecture (i.e., admin/root control).

This is the **key place where zero trust can fail.**

REALITY CHECK

The security of a zero trust system is completely tied to the security of its authentication system; after all, a key tenet of zero trust is that the user must authenticate every time they wish to gain access to a resource.

But enormous risks are created when the authentication system is a single point of compromise. If the bad actor compromises the authentication system, they quietly gain the ability to access any part of the infrastructure.

Single Points of Compromise in the News

Below are four examples of **major newsworthy breaches** from the past few years, where an attack involved a breach of a single point of compromise. Each of these organizations put their credentials in one place or invited a privileged third party into their architecture, creating **major security risks**. Once the attacker breached these single points of compromise, valuable resources were compromised all at once.

In September 2019, hackers gained unauthorized access to the SolarWinds network. This access allowed them to pivot the single point of compromise inside the SolarWinds network, the Microsoft Active Directory Federated Services (ADFS). Once they compromised ADFS, they had the ability to issue credentials to any system in the network and move laterally to anywhere they wanted. Sensitive data, from US government agencies to Fortune 500 companies, was stolen, which had a major reputational impact on SolarWinds. Their **stock price dropped 50%** following this infamous breach, and they are still on the difficult road to recovery.

Another example of an organization being breached through a single point of compromise is the case of global shipping giant Maersk. Maersk was initially compromised via a version of Ukrainian tax software. Once the attackers got in, they pivoted to stealing the highly privileged admin credentials used to manage the computers at Maersk. Because the same admin credential was widely used across machines in the entire company, this admin credential was a single point of compromise. The attackers used this admin credential to take over other computers on the network. This led to a multi-week disruption to the global shipping industry, taking the company offline for weeks. The CEO later stated that the breach cost the company over **$250 million.**

The Colonial Pipeline hack of 2021 started with a stolen VPN password and continued when hackers breached Colonial Pipeline's identity provider. Again, this is an example of an identity provider creating a single point of compromise that allows attackers to move laterally within the network, often undetected. It led to Colonial Pipeline shutting down its pipeline system for several days, causing **gas shortages throughout the US.** Failures like these are one of the reasons why the US federal government is deprecating VPNs in favor of a zero trust security posture, where users are required to authenticate every time they wish to access a resource.

Most recently, hackers breached Uber by first compromising an employee credential, and from there, pivoting toward owning Uber's privileged access management (PAM) system. A PAM stores admin credentials to other systems. This PAM was a single point of compromise; once it was hacked, the attacker was able to gain admin credentials to other sensitive and critical Uber systems. When the news broke, **Uber's stock price dropped by 5%,** and the reputational impact has yet to be fully understood.

These examples demonstrate that an organization needs to be careful about creating a single point of compromise that can be exploited by attackers. Even though these single points of compromise have been hacked time and time again, many organizations continue to rely on this practice, even when building out architectures that are "supposedly zero trust."

To return to the locks-on-doors analogy, zero trust is like locking the door at night. It feels good, and it makes us feel secure. But we don't actually know if it works until someone kicks the door in. The real test is whether an attacker can take over a single point of compromise once they kick the door in. Because if they can, they can **silently compromise the entire system.**

Application Privilege

Every one of these attacks we just discussed was so devastating because they involved a single point of compromise. Almost every access solution or security architecture, such as PAM, bastion hosts, SSO providers or certificate authorities can become a single point of compromise.

Identifying and breaching the single point of compromise, where the most **sensitive credentials** necessary for privilege escalation are held (the second phase of an attack), is where most of the damage is done. Most organizations are concerned with the first phase—managing a user's initial access—and they neglect the second phase, privilege escalation. However, examples of major breaches over the past few years demonstrate the danger of privilege escalation and lateral movement.

The modern approach to solving this problem centers on **eliminating long-lived credentials.** This is the approach of zero trust. But eliminating long-lived credentials is not a silver bullet, because attackers can still get in. With Maersk, for example, the attacker kicked in the door by exploiting Ukrainian tax software. With SolarWinds, they kicked in the door by exploiting the Orion software that was running on victims' machines. If an organization's zero trust strategy involves setting up an all-powerful authentication system that is a single point of compromise, then that organization should be ready to face **significant risks** if an attacker ever does kick in a door.

There is a better way—**trustlessness.** With a trustless architecture, access is granted without creating a single point of compromise. Eliminating single points of compromise limits the risk that a single breach of an organization then leads to a wholesale compromise of all of its systems.

In what follows, we'll walk through the key pillars of a **trustless access architecture.** A trustless access architecture eliminates single points of compromise, as defined by the SANS Institute, and meets the requirements of NIST Special Publication 800–207 on zero trust.

The Need for Trustless Access

Organizations have more risks and privileges to juggle than their people, processes, and technology can manage.

Trustless access solves this problem.

Defining Trustless Access

Security breaches seem to become more commonplace as each day passes. Board members and investors want reassurance that steps are being taken to control or minimize the damage when the inevitable happens. However, it is worthwhile to take another look at the possibility of actually eliminating the requisite conditions for a breach.

As we've said, "zero trust" is a bit of a misnomer. It actually requires a lot of trust to be placed in systems that manage authentication, secrets, credentials, and access, which create a single point of compromise. But **today's threat landscape demands that trust be minimized** and single points of compromise be eliminated. This is where trustless access comes in.

But how exactly is trustless access different from zero trust?

Trustless access consolidates the essential components of access that cloud-native organizations need for zero trust access **without introducing a single point of compromise. It also reduces the attack surface and limits risk.** This frees organizations of the need to rely on trusted third parties or over-privileged tools that have the power to control or grant access to important resources.

If organizations want a trustless architecture, they first need trustless access.

The Key Pillars of Trustless Access

At its core, trustless access is based on **five fundamental pillars:** multi-root authentication, passwordless credentials, centralized policy, identity-aware logging, and zero entitlements.

Multi-Root Authentication

Simply stated, multi-root authentication ensures that the user authenticates themselves to **multiple independent roots of trust.** Rather than just authenticating to just their SSO provider (i.e., Okta) as a single root of trust, the user additionally authenticates to another root of trust. By having the user authenticate to more than one root of trust, we ensure that the authentication system does not become a single point of compromise. Instead, an adversary would need to compromise multiple roots of trust to compromise the system.

Passwordless Credentials

In a world where everything needs a password, protecting different long-lived secrets is difficult and risky. IT and security should not need to give users passwords to individual systems and targets. This **lowers operational overhead** for IT admins, and users don't have to worry about remembering them or accidentally exposing them.

Centralized Policy

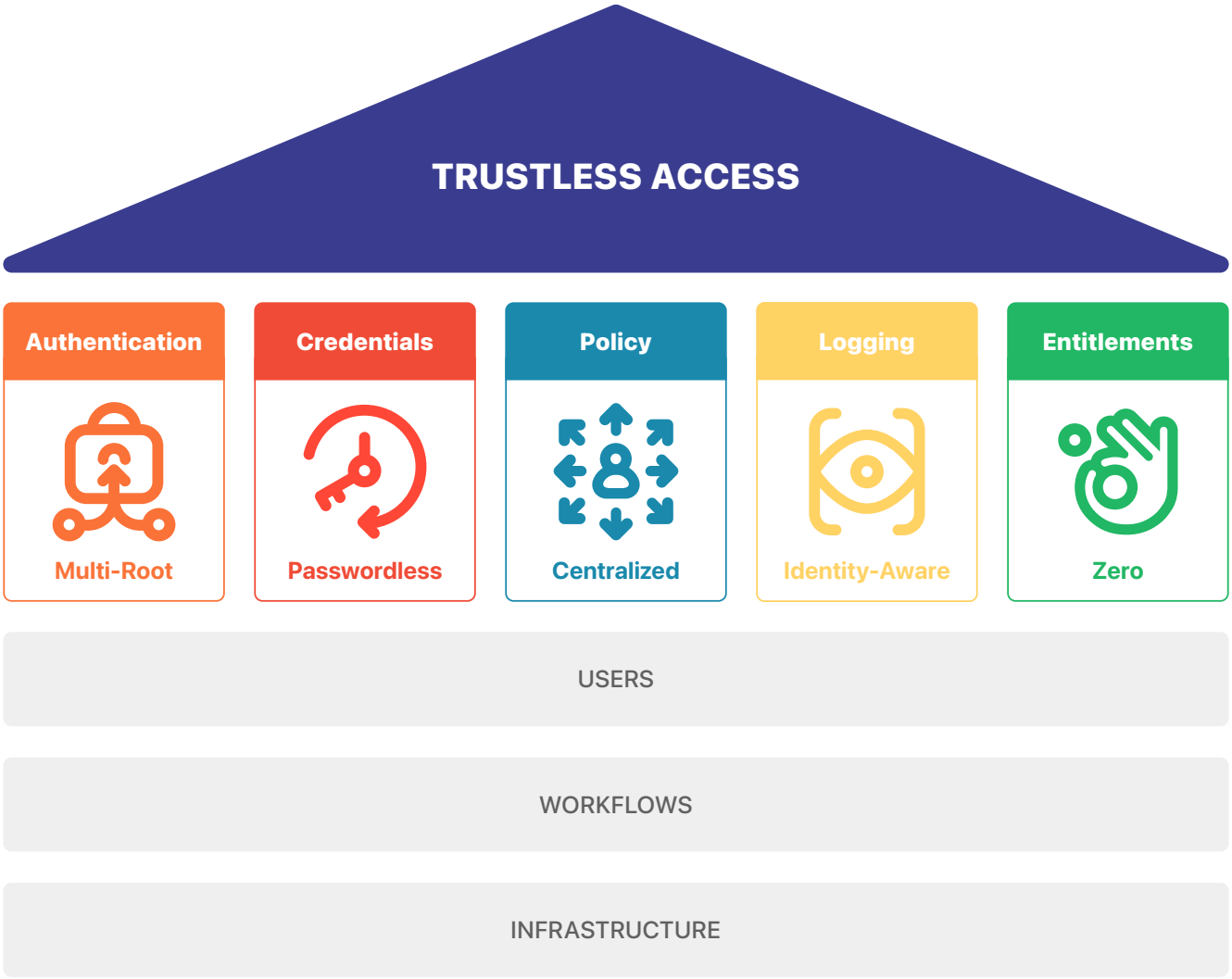
Policy management can feel like a juggling act when IT teams need to keep track of who has access to what systems and targets, when, and for how long. Trustless access requires policies to be centralized, **allowing organizations to control access to their targets via a single web console or API endpoint.** This makes it possible to enforce the principles of least-privilege access and control exactly which user can assume which role/account on which target, across all clouds and environments.

Identity-Aware Logging

Understanding who accessed which role/account on a target, along with what they did to the target shouldn't feel like guessing in the game of Clue (it was Professor Plum in the conservatory with the candlestick!). To achieve trustless access, an organization must have **centralized, identity-aware logging** (it was Professor Plum logging in as root in server nyc2-prod-xyz!). This approach supports an organization's forensics and compliance requirements.

Zero Entitlements

The access solution itself should not be a single point of compromise and **should not require privileged access** to an architecture to function properly. Such entitlements are also referred to as "authorizations," "privileges," "access rights," "permissions," and/or "rules" across platforms, applications, network components, and devices. In this guide, we've mostly referred to this concept as privileged access.



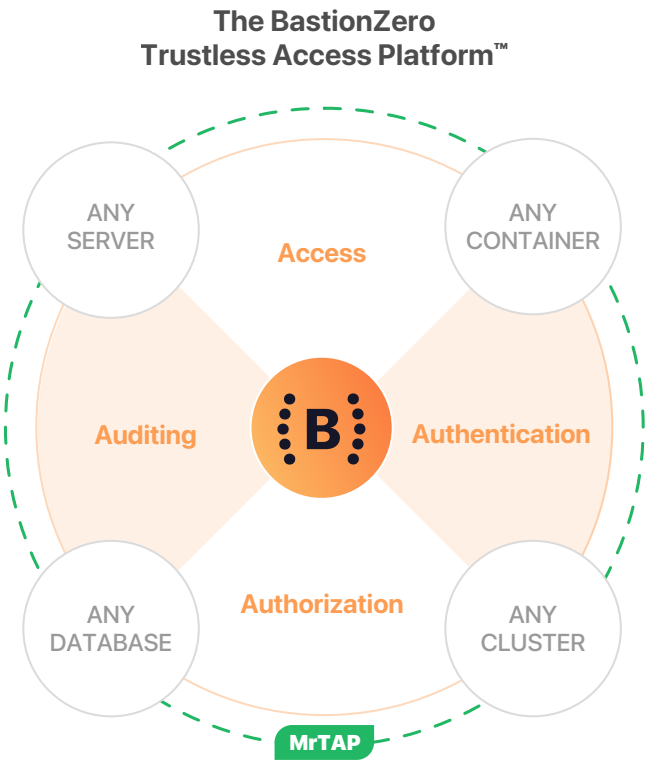
In addition to these five essential pillars, teams should look for a trustless solution that is **modern, agile, scalable, and transparent**. A modern solution must be **passwordless** to bolster security. Agility is key: It gives users just-in-time access and works across multiple workflows to boost productivity. For scalability, improved admin productivity comes from simple, easy-to-read policy controls that focus on determining who has access to which targets rather than on networks or IP addresses. Finally, transparency is crucial, including identity-aware logging, for audit and security investigations.

Ultimately, trustlessness, as a quality of both the solution's architecture and the architecture it enables, should be **the acid test used to measure resilience.**

It should come down to one very simple question:

Does a compromise of your access system result in unfettered access to and control of your users, workflows, or infrastructure?

Discover BastionZero



BZ Access

Connect your team to infrastructure resources securely, wherever they are.

BZ Authentication

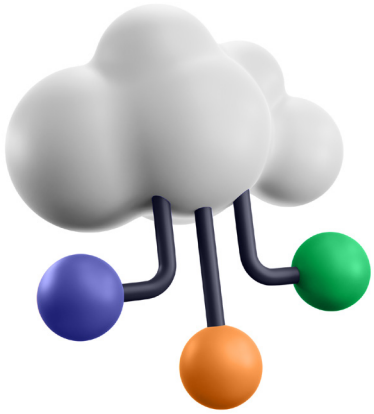
Authenticate users and services using your SSO and our independent MFA.

BZ Authorization

Enforce powerful policies based on roles and user accounts via a single pane of glass.

BZ Auditing

Memorialize all activity in your systems with command logs and session recordings.



BastionZero enables trustless access and requires no additional infrastructure to deploy or manage.

BastionZero is the first and only **cloud-native solution** for trustless access. Unlike other solutions, BastionZero **provides multi-root authentication while maintaining zero entitlements** to your systems.

BastionZero's Trustless Access Platform connects teams to resources **without risking the keys to your kingdom.** With our solution, you can **reclaim your architecture** from over-privileged third parties and ensure that the right people have access to the right resources at just the right time—every time.

> **MrTAP**
> **Multi-Root Trustless Access Protocol**
>
> BastionZero ends over-privilege and single points of compromise with MrTAP, a cryptographic protocol that certifies every request using two signatures from independent roots of trust: our cloud service (MFA) and your single-sign on (SSO) or identity provider (IdP).

By using the BastionZero platform, your organization can effectively **eliminate single points of compromise** in your infrastructure. BastionZero combines trustless access with cloud-scale efficiency—all while allowing your organization to **unlock next-level productivity with 360° visibility.**

IT and security teams can trust less, while everyone else can access more.

How Does It Work?

BastionZero's groundbreaking technology helps organizations fully realize the security and productivity that comes from trustless access. Furthermore, BastionZero is the only solution that provides **all the essential components needed for trustless access.**

The first is multi-root authentication. BastionZero removes single points of compromise with MrTAP, a cryptographic protocol that certifies every request using two signatures from independent roots of trust: our cloud service and your SSO or IdP.

Splitting authentication between two independent roots of trust ensures that your authentication system does not become a single point of compromise. Better yet, neither root of trust has unilateral entitlements to your architecture.

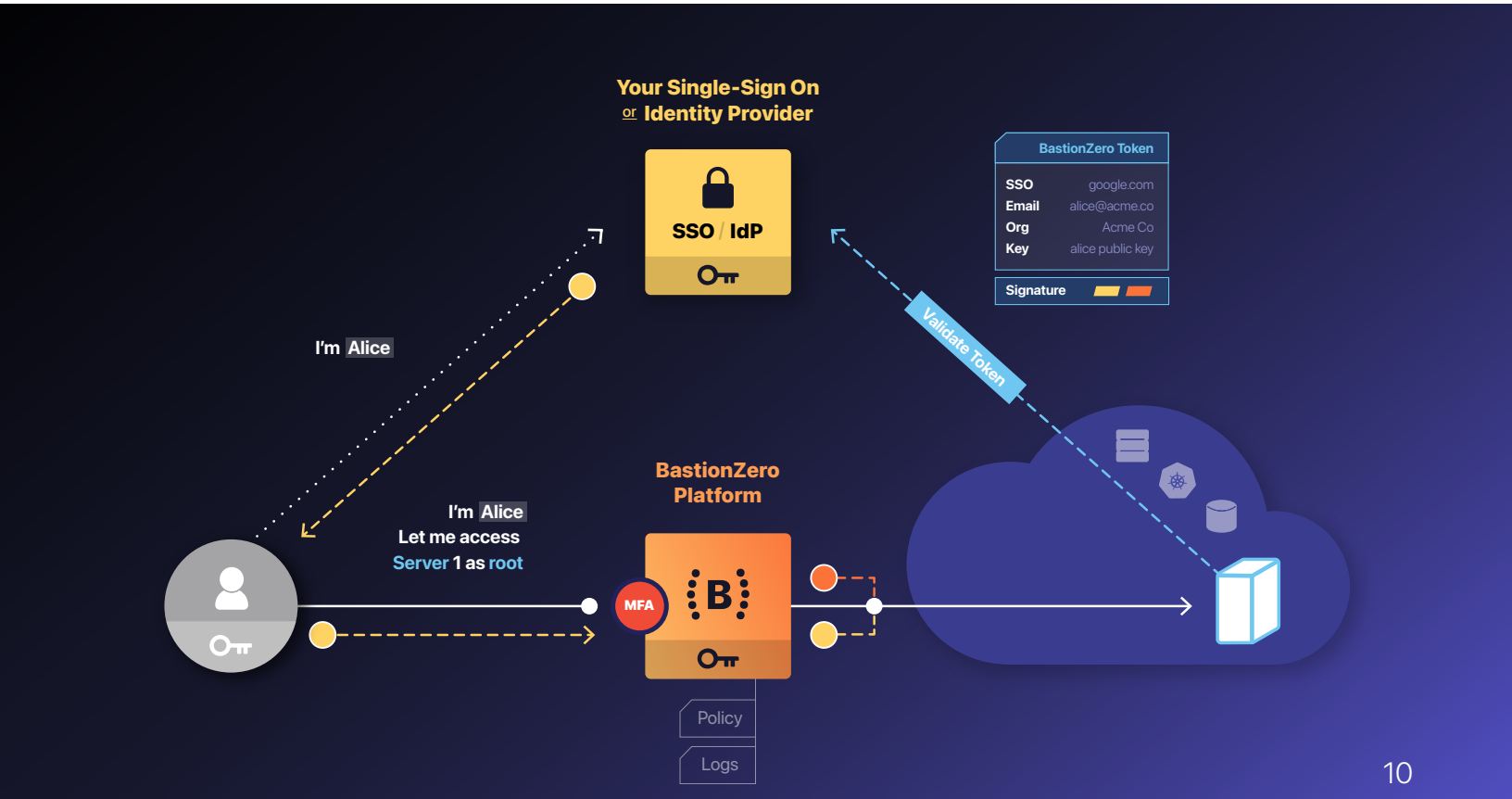
BastionZero also **eliminates passwords and long-lived credentials.** BastionZero does not store credentials to your infrastructure in a risky secrets vault (remember the Uber incident?) because our truly passwordless approach eliminates the need for long-lived credentials. Whenever credentials are required (e.g., when granting access to a database)

BastionZero uses a decentralized authentication mechanism **powered by short-lived certificates** that exist for only as long as they are needed to authenticate and authorize privileged connections.

What is unique about the BastionZero approach is that these short-lived credentials are computed by **a *decentralized* set of authorities.** This means that multiple independent parties need to cooperate to create a short-lived credential. This ensures that single points of compromise are not introduced and prevents the BastionZero platform from gaining privileged access to your infrastructure.

Remarkably, BastionZero requires **zero entitlements.** Although other tools may be able to recreate qualities associated with trustless access, the issue of entitlements is where they fall short. All of these so-called solutions need root privileges, whereas BastionZero requires **none**. The risk of unchecked entitlements in your infrastructure is clear.

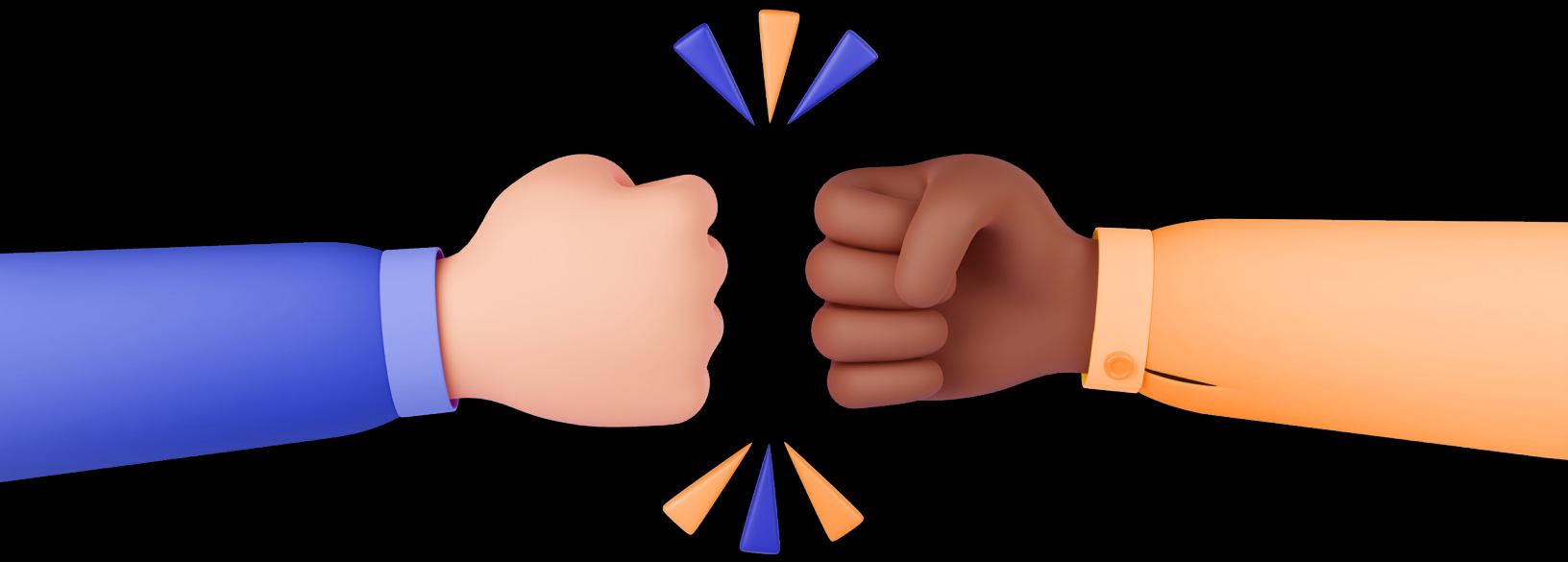
Ultimately, BastionZero helps you **trust less and access more** to keep your architecture secure.



Bastion zero

Start balancing security
with productivity **now**

Try BastionZero



BASTIONZERO.COM