How To CISO

Zero Trust Principles

A How to CISO Handbook

In the 2010s, the cybersecurity community was introduced to the concept of *zero trust*, the idea that implicitly trusting remote systems might be a … bad idea. John Kindervagt coined the term while at Forrester Research, although practical applications were developed in parallel elsewhere. In response to the breaches from Operation Aurora, Google implemented its BeyondCorp architecture and Akamai developed its Enterprise Application Access. Quickly, Zero Trust Network Access became a market space to transform corporate IT networks as vendors rushed to implement the tenets of zero trust. Core to these architectures was the understanding that the access models that had dominated IT networks for the past few decades were insufficient to protect an enterprise, and that new models needed to be deployed to protect critical infrastructure.

The Zero Trust Architecture

At the core of the new wave of zero trust architectures built in the 2010s was the elimination of *implied trust based on network location*. Rather than trusting a system because it had a user's password (maybe) and was coming from an IP address within a private corporate network, applications would need to carefully authenticate each user, relying on capabilities that had matured enough to be relied on.

TLS, the successor to SSL, was in widespread availability for servers, which allowed end user systems to more reliably authenticate that an application server was actually the expected application. At the same time, user systems could be provisioned with TLS certificates of their own, providing stronger authentication to servers than an end-user was coming from a known system associated with them.

Multi-factor authentication (MFA) systems became easier to use, enabling enterprises to rely less on passwords, and more on authenticating that a user held a token (or smartphone). Push-based to provide out-of-band authentication challenges, applications that generated time-based tokens, and security keys with cryptographic authenticators all came into widespread use.

Applications could now *authenticate* user access in a way that they couldn't before, and companies began to move from network-based VPNs to remote application access. While servers now had access to user information post-authentication, not all applications were designed to make authorization management easy to tightly configure, and *zero trust* became less about implementing the principle of least privilege and more about implementing strong authentication over the network.

Defining Zero Trust Principles

Despite an enthusiastic start – one year at the RSA Conference, it seemed like every vendor inserted *Zero Trust* into their pitch, and a wave of founders seeking investment claimed to be *Zero Trust* anything[1], Zero Trust seems to have fallen by the wayside. That doesn't mean we can't take a Zero Trust approach to planning for future architectures. But to do so, we must first define the core principles that we'll follow.

Individualized, strong authentication. All entities should be able to verify their identity remotely, in a fashion that uniquely identifies them, and is hard to steal. In practice, this requires that we plan for a world that eliminates persistent passwords: once used to authenticate one entity to another entity, the second entity can pretend to be the first entity to other entities.

Limited assumption of privilege. One entity should find it challenging to pretend to be another entity, even within an administrative scope. This is obviously imperfect – an application that operates inside another application's space will have a hard time protecting its privileges from an adversarial host – but we should aim to limit the power of adversarial administrative capabilities.

Minimized unused privilege. Entities with permissions that they do not use represent a significant risk to enterprises, and permissions should be tightly coupled to actual tasks. This is not as easy as it sounds, because most organizations don't *know* what their employees ought to be doing.

Isn't this just Least Privilege by another name?

Yes … and no. Least privilege has been a concept that has percolated through the cybersecurity industry since long before the preface "cyber" meant "computers." At a simple conceptual level, the principle of least privilege is an *authorization* goal: give no entity more access rights than it requires to accomplish its business needs. In practice, least privilege was implemented primarily through management of access rights, with a focus on eliminating as many as possible. Generations of information security professionals were educated on various models to define access rights, focusing on either integrity (Biba) or confidentiality (*), with models from pure access lists to role-based access control. The difficulty of implementing robust authorization models in modern enterprises, as users move from role to role in muddy transitions have often made least privilege into a phrase honored more in the breach than as a core rule.

While authentication improvements would sometimes come into play, it's a fair simplification to assert that the past 40 years of least privilege accomplished little beyond occasional access rights reductions and frequent reviews to satisfy compliance objectives.

---

[1]Our favorite might have been the "Zero Trust Blockchain for Big Data" pitch – Zero Trust hasn't entirely revolutionized the industry (although strong authentication has certainly taken off).