# All.Net Analyst Report and Newsletter

***Welcome to our Analyst Report and Newsletter***

## Don’t trust Zero Trust or so-called Best Practices

Until recently, the concept of zero trust architecture as a cybersecurity approach was a minor thorn in the side of those of us trying to achieve effective risk management in cyber systems. Similarly, the so-called “best practices” approach to cyber security that I have heard about for years has produced no such “best” anything. The best we can do is to try for reasonable and prudent (as apposed to unreasonable and imprudent) practices. But that has now changed with the Executive Order in which the President of the United States has declared that “The Federal Government must adopt security best practices; advance toward Zero Trust Architecture; ”

## Congratulations – Cybersecurity is reaching the heights of hype of AI

Yes, just as we seem to believe that computers are somehow smart enough to instantly reconstruct a crime scene and generate high resolution details out of a low resolution digital picture, we now have presidential mandates for the US Federal government to achieve the hyperbole of cybsesecurity.

## So what is ZTA and how is it hype?

According to NIST SP800-207:

> *Zero trust (ZT) is the term for an evolving set of cybersecurity paradigms that move defenses from static, network-based perimeters to focus on users, assets, and resources. A zero trust architecture (ZTA) uses zero trust principles to plan industrial and enterprise infrastructure and workflows. Zero trust assumes there is no implicit trust granted to assets or user accounts based solely on their physical or network location (i.e., local area networks versus the internet) or based on asset ownership (enterprise or personally owned). Authentication and authorization (both subject and device) are discrete functions performed before a session to an enterprise resource is established. Zero trust is a response to enterprise network trends that include remote users, bring your own device (BYOD), and cloud-based assets that are not located within an enterprise-owned network boundary. Zero trust focuses on protecting resources (assets, services, workflows, network accounts, etc.), not network segments, as the network location is no longer seen as the prime component to the security posture of the resource. This document contains an abstract definition of zero trust architecture (ZTA) and gives general deployment models and use cases where zero trust could improve an enterprise’s overall information technology security posture.*

Let’s pick that apart:

- ZT has nothing apparently whatsoever to do with not trusting things. It deems that location, whether physical or network, or ownership, are not to be used as a basis for trust. Rather, authentication and authorization (not identification apparently) are to be used as a basis for trusting sessions (sequences of communications) between components (including people).

  

But adopting this approach means that physical security is not a basis for trust. This is insanity of course. Without physical protection, we cannot count on anything being what it is. If I steal, forge, guess, or leverage an authentication, I will be authorized to do whatever it allows. It also is inconsistent with (and apparently exempts) all of the classified systems of the US Federal government, so I guess we don't trust it for the really important stuff. Per SP800-207 "It is meant to aid understanding of zero trust **for civilian unclassified systems**..."

- ZT is a response to a trend for low-consequence uses from afar in industry.

Great! There are other trends, like lying in large-scale Internet forums. Should we than adopt a standard of security assuming everything is a lie in such forums? Throw out the baby with the bathwater so to speak? ZT seems to say that we should adopt the misnomer of "Zero Trust" and turn it into a meme in society that says we can somehow not trust anything and still be successful. But adoption of misleading propagandistic hyperbole most importantly tends to lead to misunderstandings, particularly by executives (apparently at the highest level) that current notions are things to be adopted as a universal approach.

The tenants of ZTA are:

- All data sources and computing services are considered resources.
  - *Sort of… for example, "the enterprise may choose to classify personally owned devices as resources…" - so All doesn't mean all – it means the ones you choose. And what exactly is a source? I suspect they mean a source of network sessions, because that's what they seem to think should be authenticated. But the source of the content and its provenance are more complicated than this because of the transitive flow of information. For example, the source of the Solar Winds attack was Solar Winds, which apparently authenticated its sessions for distribution (and the content).*
- All communication is secured regardless of network location.
  - *This is ridiculous of course. Within devices, due to physical proximity (i.e., network location), we are not going to encrypt all communications, not can we encrypt all communications, because the protocols themselves are not encrypted at the highest level. The Internet will not work if the routers and switches don't have access to the IP address and other header fields required to route traffic. And encrypting it all en route will require a key management infrastructure that does not yet exist. Now I agree with the SP that internal communications should be encrypted for the most part, but I don't think that all content on internal and external Web sited should have mandatory encryption. This would require that there be no more http – ONLY https. And it would require that console traffic be encrypted, which does not work on most existing systems. And it would require encryption between control mechanisms and the OT they control on direct wiring. Pure insanity, especially when the encryption will slow the loops for OT processing to the point where they are no longer effective at control. The additional statement "All communication should be done in the most secure manner available, protect confidentiality and integrity, and provide source authentication." is also ridiculous. The most secure manner available is typically far more expensive than a reasonable and prudent approach, and what about transparency, use control, and*

  

*other requirements of these systems. Also note that forcing end-to-end encryption of all traffic eliminates network-based anomaly and intrusion detection in most cases. And that means that only endpoints can do detection, which increases their workload and forces more traffic for correlation if correlative methods are desired.*

- Access to individual enterprise resources is granted on a per-session basis.
  - *This may be great for session-based mechanisms, but plenty of mechanisms don't use sessions. For example, DNS lookups over UDP are not session-based, nor is ICMP, nor DHCP, or many other protocols. Furthermore, these are not typically authenticated at all, and are the means by which a further authentication process may be started. And then there are things like streaming services which are multicast and session requirements on multicast could easily break the entire model they use. And what is an "individual enterprise resource" exactly? Is it a database or a record in that database or the system containing the database?*
- 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.
  - *So the policy has to change with time… but I don't think they really mean policy by this, but rather technical access control policy or some such thing. Decisions based on the policies of the entity, not changes to the policies themselves. This is also a very problematic area. I will stick to the things they have required for the moment, but rest assured the rest is at least as problematic.*
  - *"For zero trust, client identity can include the user account (or service identity) and any associated attributes assigned by the enterprise to that account or artifacts to authenticate automated tasks. …" So identity is not about people, and thus we have broken the part about who is using the resource. So if I steal your computer, I am OK to use the mechanisms of zero trust architectures as if I were you.*
  - *"For zero trust, ...Requesting asset state can include device characteristics such as software versions installed, network location, time/date of request, previously observed behavior, and installed credentials." Of course 'can include' includes 'does not include', so this is relatively meaningless. Nevertheless, software versions installed would presumably require that each session start with the SBOM and DBOM inventory of the mechanism being dumped to the other side of the session, and of course since encryption is required and this is part of authentication, we need some sort of bootstrap process for this. We could simply provide an SBOM/DBOM identity if desired, but then that would allow erroneous SBOM/DBOM content or reuse from another mechanism (just copy the IDs and you are good to go). Apparently time and date are acceptable while location is not. And here we are allowed to use network location, despite being told earlier that ZT is about NOT trusting based on network location. Confusing isn't it?*
  - *There is more problematic stuff in this area, like "These rules and attributes are based on the needs of the business process and acceptable level of risk. Resource access and action permission policies can vary based on ... Least privilege, ..." But I will move on to save your time and patience.*

  

- The enterprise monitors and measures the integrity and security posture of all owned and associated assets.
  - *What exactly associated assets are is not clear, but I suspect it means everything you are authenticating. So the question is how you "monitor and measure the integrity" of things you do not own and do not control. "No asset is inherently trusted." - sounds right, but then how do we bootstrap trust? If we don't trust any asset, what do we trust for what purpose and to what extent to act as the basis for creating the trust metrics?*
  - *"An enterprise implementing a ZTA should establish a continuous diagnostics and mitigation (CDM) or similar system to monitor the state of devices and applications and should apply patches/fixes as needed." Sure – but how to do this for 3rd party assets you don't own? Especially when the other party cannot trust your assets to act appropriately in their system without trusting yours. So every system must allow MUTUAL verification, which means I need to be able to look inside your systems to the same extent as you need to be able to look into mine. Sounds to me like a transparency requirement… but transparency is not part of the ZTA apparently. And what enterprise would allow me such access in order to hire me as a worker or form a business relationship with my company? The current methodology is that a 3rd part audit was done at some point in time.*
  - *There is more generic maybe and might stuff here, but I will ignore them for now.*
- All resource authentication and authorization are dynamic and strictly enforced before access is allowed.
  - *Note the term "before", not during. But fear not, "This is a constant cycle of obtaining access, scanning and assessing threats, adapting, and continually reevaluating trust in ongoing communication." So we now need a whole new technology for continuous monitoring of all of the internals of every system connected to each other system. More software, more embedded functions, more chances for more bugs in more things required in order to establish trust, taking more resources than most of the actual activities being done by the systems.*
  - *"An enterprise implementing a ZTA would be expected to have Identity, Credential, and Access Management (ICAM) and asset management systems in place." So we now need the basis for trust in all of these systems and vendors that also has to be continuously monitored and adapted. And if we should end up with reduced trust in one of these systems, that means the whole network might have to stop working till we mitigate that loss of trust.*
  - *Some other things are listed here, but again, I will save you the time.*
- The enterprise collects as much information as possible about the current state of assets, network infrastructure and communications and uses it to improve its security posture.
  - *As much as possible? Pure insanity! It is possible to collect far more than can ever be used, and of course there are those pesky requirements for confidentiality that have to be attended to in building the ultimate monitoring system. Of course because all traffic is encrypted, this means sensors to collect everything at*

  

*endpoints and then send it back to the collection system, which more than doubles all traffic requirements for every communication. Massive storage arrays, and if it is to be used, massive compute power.*

**An example?**

It turns out there are no actual examples. Zero Trust is just a concept. And implementation of that concept is now mandated be executive order. No resources behind it – just an order.

But they do give a "zero trust view of a network"

- The entire enterprise private network is not considered an implicit trust zone.
  - *Network zoning has existed for more than 20 years, with DMZs and firewalls around for more like 30 years. This is nothing new.*
- Devices on the network may not be owned or configurable by the enterprise.
  - *This (may in the context of perhaps, not prohibition) too has long been true of most enterprise networks, even if not explicit in the architecture in some cases.*
- No resource is inherently trusted.
  - *This too has always been true – in the sense that trust does not work that way. Inherent trust without limit is an oxymoron because trust is a decision, and as such, it can go more than one way and change with time. Of course the definition of trust (usually something like the extent to which you are willing to suffer harm by the, in this case 'resource') includes inherent potential for loss by anything and everything. But the rating of loss in terms of consequences underlies this concept.*
- Not all enterprise resources are on enterprise-owned infrastructure.
  - *A definitional question here – what constitutes an "enterprise resource"? And what is "on" and what is "enterprise-owned"? If your enterprise uses the Internet, telephony, the road system, external sources of energy, or external law enforcement resources (etc.) then these things are not enterprise owned, and you can bet your information and mechanisms are "on" (or depend on) these things from time to time. So this is pretty much ridiculous as a differentiating criterion for almost any enterprise.*
- Remote enterprise subjects and assets cannot fully trust their local network connection.
  - *Subjects – a new term here – referring presumably back to the subject object model of security. Resources vs. subject and assets… what are they, how do we tell them apart? And what is a "remote enterprise subject" (or asset). Remote implies some sort of central I think, and that would seem to imply a physicality. But ZTA is independent of physical location. And since they cannot trust ANY of their network connections (or networks) until authenticated under ZTA, how is this any different from their non-local network connection. This is just gobbledygoop. And what is "fully trust" anyway? Zero trust would seem to mean not trust at all, but of course means no such thing. Fully trust is also meaningless.*

  

- Assets and workflows moving between enterprise and nonenterprise infrastructure should have a consistent security policy and posture.
    - *Of course this is not attainable in the reality of cloud-based systems, at least not taking advantage of the efficiencies of such systems. And since when does a workflow move? And in what sense do assets and workflows have security policy and posture? These are huge expressions of vast things asserted in a generic way without meaningful utility.*

*But then this is consistent with the title of the section in SP800-207 – "A Zero Trust View of a Network". In essence, in a view of the ill-defined and under-specified conceptual architecture called Zero Trust, consistency, sensibility, and meaningful statements are not required.*

**Logical components of the architecture**

In section 3, SP800-207 identifies a set of components (not apparently resources) that make up a deployment. "These components may be operated as an on-premises service or through a cloud-based service." - I thought that location independence for resources was inherent in a ZTA, but apparently this had to be made explicit. "Note that this is an ideal model showing logical components and their interactions." The components are seemingly derived from the identity management components of the Burton Group architecture from the early 2000s, but NIST has a habit of no referencing sources in their SPs, so Dan Blum should be given credit here:

- Policy engine (PE):
    - *This is where the magic happens. Somehow, this engine takes information from the Policy administrator and turns it into a constant stream of ever changing decisions regarding trust and authentication requirements. If this sounds like the AI part, it is. Of course there are real implementations of this in rule-based systems from the late 1990s and early 2000s by IBM and others. But the level of flexibility and complexity here makes it problematic for this implementation, and all of this requires a high level of inherent trust in the technical security policy, the human specification mechanism, and the engine.*
- Policy administrator (PA):
    - *The policy administrator generates all the tokens for authentication and is thus of course also highly trusted because it implements the decisions. So this is a highly trusted component of the architecture that, if it fails, causes potential harm effecting the entire enterprise (and it's supply and demand chain).*
- Policy enforcement point (PEP):
    - *"Beyond the PEP is the trust zone (see Section 2) hosting the enterprise resource.". OK, this is what seems like the "Trusted Zone" apparently from the Burton Group architecture again, and in this case, I am the one who wrote that part of the fully layered architecture (including the control and audit zones) along with the rest of the BG team.*

I could go on, but it only gets more ridiculous. The so-called Zero Trust approach apparently has a centralized set of mechanisms requiring massive trust. And even if physically and logically distributed, this does not solve the problem.

  

### Who shall watch the watchers?

The architecture actually includes lots of other stuff, like public key infrastructure, identity management, a SIEM system, threat intelligence, activity logs, and so forth. All of these are of course trusted in one form or another, as is the entire basis in cryptography of the entire system, and of course this high a reliance on a mechanism that is not worthy of the level of trust already placed upon it is problematic in the extreme. As an aside, I should mention that it looks like quantum computing will soon be (if not already in classified systems) able to defeat essentially all current cryptographic systems. So removing all the trust based on everything else and placing it on the fragile and brittle component of cryptography does not seem like a good move.

Zero trust architecture trusts lots of things, and they aren't necessarily the right things to trust.

### And the variations

So naturally, there are lots of variations on the theme of ZTA. Most of them are nothing like ZTA and only partial sort of implementations even less well spelled out that the overall approach. Don't get me wrong. Lots of these older longstanding ideas are worthwhile for select situations. But none of them are Zero Trust or anything like it.

They seem to have in common a separation between a logical control plane and data plane. But of course there is no such separation in a ZTA, it's just a matter of whether you can authenticate to one or the other, and they have 2-way communications, which means if we add general purpose computation anywhere we are subject to computer viruses and similar threats. And of course denial of services in the data plane (since it is logically separated only) means potential denial of services in the control plane, which puts the whole architecture at risk from outages in the policy decision process. But then service availability was not one of the specified objectives of ZTA, so shutting down a major supply of fuel is not covered by the ZTA approach. Which is to say, the President put out an EO that in this aspect doesn't even address the critical situation that drove the rapid (perhaps premature) deployment of the EO.

### This is how it often works

I've seen this movie. A major failure happens. The CEO says to the folks who work for them, what can I do right away to fix this. They each come up with the same list of ideas they have been pushing for a while, and the list becomes the decision which then comes out in an Executive Order (or whatever decision is made). The result is inefficient, inappropriate, ineffective solutions that do not either address the current need or meet the long term decision criteria for the long-term need. They get the people making the suggestions what they have wanted for years – resources. But they are not well thought out resources suited to actually help the situation. Some of them might be of course… but even a stopped clock is right twice a day (12-hour with hands).

Why should a Federal government be any different?

### The trust algorithm

Of course in order to sort out all of these trust decision, we need a trust algorithm, and we need to trust that algorithm to a level commensurate with all of the consequences of all the decisions it will make for us. So Zero Trust is really nearly Infinite Trust in the algorithm(s) making the decisions. So let's see what NIST says about this:

  

- "For an enterprise with a ZTA deployment, the policy engine can be thought of as the brain and the PE's trust algorithm as its primary thought process. The trust algorithm (TA) is the process used by the policy engine to ultimately grant or deny access to a resource. The policy engine takes input from multiple sources (see Section 3): the policy database with observable information about subjects, subject attributes and roles, historical subject behavior patterns, threat intelligence sources, and other metadata sources. The process can be grouped into broad categories and visualized in Figure 7." *Note Figure 7 takes a list of inputs, and makes a decision (check or x).*
  - **Access request:** a request from a subject (not a resource) for use of a (singular) resource.
    - *The request can include OS version and other such things.*
  - **Subject database:** "This is the "who" that is requesting access to a resource… he set of subjects (human and processes) of the enterprise or collaborators and a collection of subject attributes/privileges assigned" and refers to SP 800-63, -162, and NISTIR 7987.
    - *This is essentially a list of properties for each subject used somehow to make the determination of their being allowed to access the resource.*
  - **Asset database (and observable status):** "This is the database that contains the known status of each enterprise-owned (and possibly known nonenterprise/BYOD) asset (physical and virtual, to some extent). This is compared to the observable status of the asset making the request and can include OS version, software present, and its integrity, location (network location and geolocation), and patch level. Depending on the asset state compared with this database, access to assets might be restricted or denied."
    - *So apparently the Zero Trust location independent architecture requires location (network and geolocation) to make decisions. This is where we call BS yet again, but only the beginning.*
  - **Resource requirements:** "This … defines the minimal requirements for access to the resource. … These requirements should be developed by both the data custodian (i.e., those responsible for the data) and those responsible for the business processes that utilize the data (i.e., those responsible for the mission)."
    - *Lots of "may require" things, but no actual requirements. And they have to be developed by two parties who are apparently trusted to make the decisions about each resource based on the enterprise-defined decision about what to consider. So lots of trust here, in people to make good decisions.*
  - **Threat intelligence:** "This is an information feed or feeds about general threats and active malware operating on the internet. " It also includes information on the specific device behaviors. "...These feeds can be external services or internal scans…"
    - *So here again, we have to trust threat intelligence with all of its false positives and false negatives from 3rd party sources (even if we use our own code, we are to trust the intelligence), all of which might be exploited to deny services.*

  

- "The weight of importance for each data source may be a proprietary algorithm or may be configured by the enterprise. These weight values can be used to reflect the importance of the data source to an enterprise. "
  - *In other words, it's up to you.*

**And there's still more…**

The foolishness never ends. Here are some other jewels… network requirements:

- "Enterprise assets have basic network connectivity." *In other words, there is access to enterprise resources prior to establishing trust.*
- "The enterprise must be able to distinguish between what assets are owned or managed by the enterprise and the devices' current security posture. This is determined by enterprise-issued credentials and not using information that cannot be authenticated information (e.g., network MAC addresses that can be spoofed)." *But these credentials can be copied and so can also be spoofed.*
- "The enterprise can observe all network traffic." (up to Layer 7). *Which is to say, defeat end-to-end encryption, the very basis for ZTA trust in the first place.*
- "Enterprise resources should not be reachable without accessing a PEP" *But a PEP is an enterprise resource! So the architecture defeats its own principals in the most critical of applications, the decision process surrounding trust.*
- "The data plane and control plane are logically separate." *See above why this is problematic.*
- "Enterprise assets can reach the PEP component." *See not reachable above.*
- "The PEP is the only component that accesses the policy administrator as part of a business flow." *So how is this done in the Zero Trust model? Not via credentials, (see above) so how is it protected? Physically? Not under Zero Trust!*
- "Remote enterprise assets should be able to access enterprise resources without needing to traverse enterprise network infrastructure first." *So the PEP MUST be outside of the enterprise network infrastructure.*
- "The infrastructure used to support the ZTA access decision process should be made scalable to account for changes in process load." *So we need a scalable ZTA network where we don't trust the scaling mechanisms or resources provided until they authenticate, but of course since they are created by 3rd parties, the 3rd parties must be trusted to place the required credentials… not Zero Trust!*
- "Enterprise assets may not be able to reach certain PEPs due to policy or observable factors." *So contrary to the previous, we do need to authenticate before accessing a PEP.*

It has been said that "foolish consistency is the hobgoblin of little minds" [Ralph Waldo Emerson], but foolish inconsistency is the bane of cybersecurity, and those who depend on these systems sometimes literally die as a result of the foolish inconsistencies of the sorts we see here.

  

### Threats to ZTA

NIST did one thing sort of right in SP800-207. They listed (P28-31 - about 2/3 of the way through the document, which has the cognitive effect of reducing its impact on the reader), some threats (not actually threats, but called that by NIST) to ZTA:

- Subversion of ZTA Decision Process
- Denial-of-Service or Network Disruption
- Stolen Credentials/Insider Threat
- Visibility on the Network
- Storage of System and Network Information
- Reliance on Proprietary Data Formats or Solutions
- Use of Non-person Entities (NPE) in ZTA Administration

Of course the descriptions are short and minimize the impacts. For example, Denial of Services says “If an attacker disrupts or denies access to the PEP(s) or PE/PA (i.e., DoS attack or route hijack), it can adversely impact enterprise operations.” This use of primacy and recency is a commonly known method of influence operations, even if not intentional by NIST.

This apparently had the desired effect of being able to claim they said something about bad things without actually interfering with ZTA’s acceptance by top management who doesn’t read all that detailed stuff.

Other stuff in the middle includes possible interactions with other existing Federal guidance is then identified, again a short section in the middle.

### Migration to ZTA

Then we have the migration plan, which is described as “a journey rather than a wholesale replacement of infrastructure or processes.” essentially an eternal process of movement.

I love this paragraph:

- “In a greenfield approach, it would be possible to build a zero trust architecture from the ground up. Assuming the enterprise knows the applications/services and workflows that it wants to use for its operations, it can produce an architecture based on zero trust tenets for those workflows. Once the workflows are identified, the enterprise can narrow down the components needed and begin to map how the individual components interact. From that point, it is an engineering and organizational exercise in building the infrastructure and configuring the components. This may include additional organizational changes depending on how the enterprise is currently set up and operating. “

Nothing to it. It’s all just an engineering and organizational exercise after you know all of your workflows, applications, and services. Of course these change at a pace far faster than engineering and organizations move, so you will never catch up, and of course this ignores the other critical parts of the ZTA approach as identified above. But in the end, you will have a ZTA with only 7 major identified problems (listed above), all of which are not solvable in the architecture.

  

But suppose you need to migrate? Here's the outline (note that there is apparently more to do in migration to figure out what's needed than there is to do from a Green Fields approach, where you don't need to identify actors – as an example…) with some gems:

- Identify actors on the enterprise (so we need to first figure out every user that will ever use any resource. Facebook only has another 6 Billion to go… and it's roles and rules just like it was before ZTA...)
  - *"For a zero trust enterprise to operate, the PE must have knowledge of enterprise subjects. Subjects could encompass both human and possible NPEs, such as service accounts that interact with resources.*
  - *Users with special privileges, such as developers or system administrators, require additional scrutiny when being assigned attributes or roles. In many legacy security architectures, these accounts may have blanket permission to access all enterprise resources. ZTA should allow developers and administrators to have sufficient flexibility to satisfy their business requirements while using logs and audit actions to identify access behavior patterns. ZTA deployments may require administrators to satisfy a more stringent confidence level or criteria as outlined in NIST SP 800-63A, Section 5 [SP800-63A]. "*
- Identify assets owned by the enterprise (generally a good idea, but… at this level of granularity and the need to update it for example for every update to every system over time, this is likely a nightmare of cost and complexity. And they still want to use the things they told us were no longer to be used – network location, MAC address, ...)
  - *"This goes beyond simply cataloging and maintaining a database of enterprise assets. This also includes configuration management and monitoring. The ability to observe the current state of an asset is part of the process of evaluating access requests (see Section 2.1). This means that the enterprise must be able to configure, survey, and update enterprise assets, such as virtual assets and containers. This also includes both its physical (as best estimated) and network location. This information should inform the PE when making resource access decisions.*
  - *Nonenterprise-owned assets and enterprise-owned "shadow IT" should also be cataloged as well as possible. This may include whatever is visible by the enterprise (e.g., MAC address, network location) and augmented by administrator data entry. This information is not only used for access decisions (as collaborator and BYOD assets may need to contact PEPs) but also for monitoring and forensics logging by the enterprise. Shadow IT presents a special problem in that these resources are enterprise-owned but not managed like other resources. Certain ZTA approaches (mainly network-based) may even cause shadow IT components to become unusable as they may not be known and included in network access policies"*
- Identify key processes and Evaluate Risks Associated with the Executing Process (OK – so a complete process reviews at the level of every resource and use and all the workflows have to be done to engineer them into the ZTA)

  

  - "The third inventory that an agency should undertake is to identify and rank the business processes, data flows, and their relation in the missions of the agency. Business processes should inform the circumstances under which resource access requests are granted and denied. ..."

- Formulating Policies for the ZTA Candidate (more people we need to trust and mistakes that can be made with potentially catastrophic results)
  - *"The enterprise administrators then need to determine the set of criteria (if using a criteria-based TA) or confidence level weights (if using a score-based TA) for the resources used in the candidate business process. Administrators may need to adjust these criteria or values during the tuning phase. These adjustments are necessary to ensure that policies are effective but do not hinder access to resources."*
- Identifying Candidate Solutions (yes they have to be identified)
- Initial deployment and monitoring (figuring out what you really need is complicated)
  - "...*Few enterprise policy sets are complete in their first iterations: important user accounts (e.g., administrator accounts) may be denied access to resources they need or may not need all the access privileges they have been assigned..."*
- Expanding the ZTA (as long as nothing changes)
  - *"However, if a change occurs to the workflow, the operating ZT architecture needs to be reevaluated. Significate changes to the system—such as new devices, major updates to software (especially ZT logical components), and shifts in organizational structure—may result in changes to the workflow or policies. In effect, the entire process should be reconsidered with the assumption that some of the work has already been done. For example, new devices have been purchased, but no new user accounts have been created, so only the device inventory needs to be updated."*

**The simplicity principal becomes the complexity principal**

A long time ago, cyber-security had a principal called simplicity as part of the "Generally Accepted System Security Principals". The idea was that complex things are almost impossible to get right, so make it as simple as you can to avoid the complexity that kills security. This was changed a long time ago, for no particularly identifiable reason I am aware of. And it continues to come back to haunt us today.

So-called Zero Trust is really just a deception to sell an extremely complex approach to placing more trust in more things that are less worthy of trust, and putting more consequences on this system for more people.

The cure to the disease is to put the simplicity principal back in play, and start pushing back hard on high consequences that depend on and from building an even bigger house of cards. The so-called Zero Trust.

**Conclusions**

Don't trust the Zero Trust propaganda misnomer. It's hyperbole to cover a set of pre-existing methods in a so-far inconsistent, poorly thought, expensive, and time consuming approach.

  