ZTA: Never Trust, Always Verify

Char Sample[1,2], Cragin Shelton[2], Sin Ming Loo[1], Connie Justice[3], Lynette Hornung[2] and Ian Poynter[2]
[1]Boise State University, USA
[2]Independent, USA
[3]Indiana University at Purdue, USA
charsample50@gmail.com
drcragin@icloud.com
smloo@boisestate.edu
cjustice@iupui.edu
lhornung@gmail.com
ianpoynter@gmail.com

Abstract: Zero Trust Architecture (ZTA) deployments are growing in popularity, widely viewed as a solution to historical enterprise security monitoring that typically finds attackers months after they have gained system access. ZTA design incorporates multiple industry security advisories, including assuming network compromise, using robust identity management, encrypting all traffic, thwarting lateral movement, and other security best practices. Collectively, these features are designed to detect and prevent attackers from successfully persisting in the environment. These features each offer solutions to various ongoing security problems but individually are not comprehensive solutions. When designed for cloud services ZTA holds the promise of outsourcing security monitoring. However, some observations about ZTA suggest that the component solutions themselves have flaws potentially exposing systems to additional undetected vulnerabilities, providing a false sense of security. This paper addresses vulnerable paths using a bottom-to-top approach, listing problem areas and mapping them to attacker goals of *deny, deceive, disrupt, deter,* and *destroy*. The paper then addresses residual risk in the architecture. Based on the findings the paper suggests realistic countermeasures, offering insights into additional detection and mitigation techniques.

Keywords: zero trust architecture, vulnerabilities, attack, component, system

1. Introduction

Cybersecurity has a history of *magic bullet* solutions for protecting information from myriad intentional and inadvertent damaging situations (Simmonds, 2019). Highlights include bandwagons formed for encryption, discretionary access control (DAC), firewalls, virtual private networks (VPN), secure socket layer / transport layer security (SSL/TLS), public key infrastructure (PKI), blockchain, artificial intelligence (AI), et cetera. A theme in this history is recognition at each phase of the level of trust assumed for internal and external entities (humans and systems) interacting with the system of interest. Not surprisingly, a current magic bullet, zero trust architecture (ZTA) has reached the pinnacle of proposing system design and operation based on trusting no entity at any time in any situation.

Each of the technologies named above has been both a legitimate contribution to the field and a popular marketing term. Today ZTA is in the same boat, both as a design principle and a marketing term. ZTA is not a new design philosophy; instead ZTA strengthens existing trust technologies by adding additional decision points and enforcing temporal limits. The previous *defense-in-depth* approach did the same, only in smaller domains, creating stovepipes that made sharing more difficult while preserving physical security. We argue that the binary trust/distrust (or perhaps more appropriately distrust/trust) model does not reflect the complexity of work environment relationships (Campbell, 2020).

In spite of ZTA offering no new technologies, the rapid adoption of ZTA proceeds possibly because of Executive Order 14028 (Biden, 2021) which directed each U.S. federal agency leader to create plans to implement ZTA in their respective agencies. This order was part of an initiative to facilitate inter-agency information sharing while maintaining a superior security posture (Ibid). The balancing act between sharing and security is a long-standing security challenge where assumptions and implementation details create exploitable vulnerabilities. Previous design philosophies of perimeter defense and defense-in-depth emphasized need-to-know (Bell & LaPadula, 1976); ZTA prioritizes the need to share (Biden, 2021). So, the agencies may attempt to share, but individual program managers may be reluctant to do so since they are responsible for the program's security.

Adding confusion, products and services advertise ZTA as an offering, when products can at best support only component aspects of ZTA. ZTA is a set of design principles, not something that can be implemented with a single product. (Kindervag, Balaouras, & Colt, 2010). Services can also fail at ZTA by not addressing the ramifications of design decisions. Butcher (2021) noted that business requirements over time weakened perimeter defenses, giving rise to other solutions where trust is too restricted. The likelihood of business requirements undermining ZTA remains. Business goals and cybersecurity objectives require a balancing act extending beyond technical solutions into organizational behaviors, ultimately shaping the security policy that the ZTA supports.

If a site currently has a weak security posture, ZTA principles may offer an improvement but cannot assure complete security. In short, ZTA won't guarantee freedom from cybersecurity problems and in certain case ZTA could introduce additional problems. Enterprise architects need a baseline of competence in understanding technologies, underlying assumptions, and inherent security gaps, residual risk, and the operating environment before applying the ZTA.

2. Background

Kindervag, Balaouras, & Colt (2010) first defined ZTA for Forrester in 2010. Butcher (2021) presented ZTA as a *design philosophy* for security architects, a philosophy flexible enough to allow various instantiations based on site requirements. However, cybersecurity has a history of implementations varying from envisioned designs; ZTA is no exception. A quick overview of strengths and concerns associated with ZTA follows to assist in framing the discussion.

Figure 1 (Rose, et al., 2020, Figure 2) depicts the top-level overview of the NIST ZTA. This graphic shows the overall security system with input and output elements, processing functions, and the interactions between elements and functions. Examination of the system provides attack targets from the system supply chain, through hardware, software, and ultimately humans. Trust is complicated and thus a vulnerability (Campbell, 2020). Traditional mechanisms to grant trust are multi-faceted, but the trust decision is binary; this is still true in ZTA implementations. Trust is granted based on the specific transaction, the confirmed identities of participating entities (human and system), the circumstances surrounding the transaction, and time.

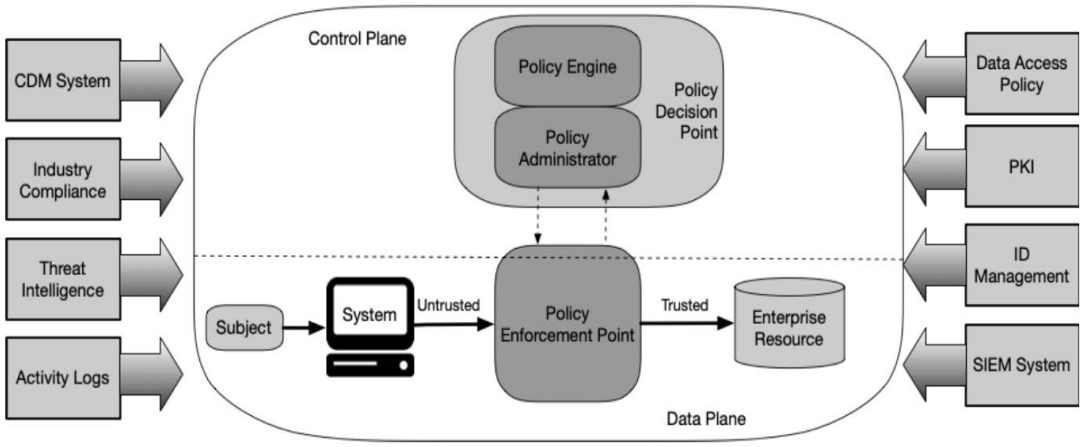

Figure 2: Core Zero Trust Logical Components

Figure 1: Zero trust architecture overview (Rose, et al., 2020, Figure 2)

2.1 Strengths

The NIST Zero Trust Architecture (Rose et al., 2020) recognized that ZTA will be an evolving approach, requiring resiliency and ongoing evaluation as technology, security threats, and protection tools change. The NIST ZTA supports security with identification, authentication, and authorization for users, assets, and resources. Also important are data and service protection to manage risks, protecting credentials and endpoints with encryption, Multi-Factor Authentication (MFA) and other mechanisms. Continuous Diagnostics and Mitigation

(CDM) is used to ensure patches and fixes are applied. The NIST Risk Management Framework (RMF) (Ross, et al., 2018), and Privacy Framework (Lefkovitz & Boeckl, 2020) are used with ZTA as needed to ensure sensitive data and certificates have appropriate encryption and other critical data controls are implemented. The Trusted Internet Connection (TIC) (Weichert, 2019) has expanded to include cloud and mobile environments, also critical elements in U.S. Federal government systems ZTA.

The use case shown in Figure 2 (NSA, 2021) illustrates ZTA with security safeguards in place, following the Cyber Kill Chain (CKC) steps (Lockheed Martin, 2015). Thus, once an adversary gains access and initiates lateral movement inside the now-compromised system, the policy enforcement point (PEP) can block that movement. When an attacker tries to impersonate a legitimate user, seamless MFA prompts are at play and automatic blocking operates as intended.

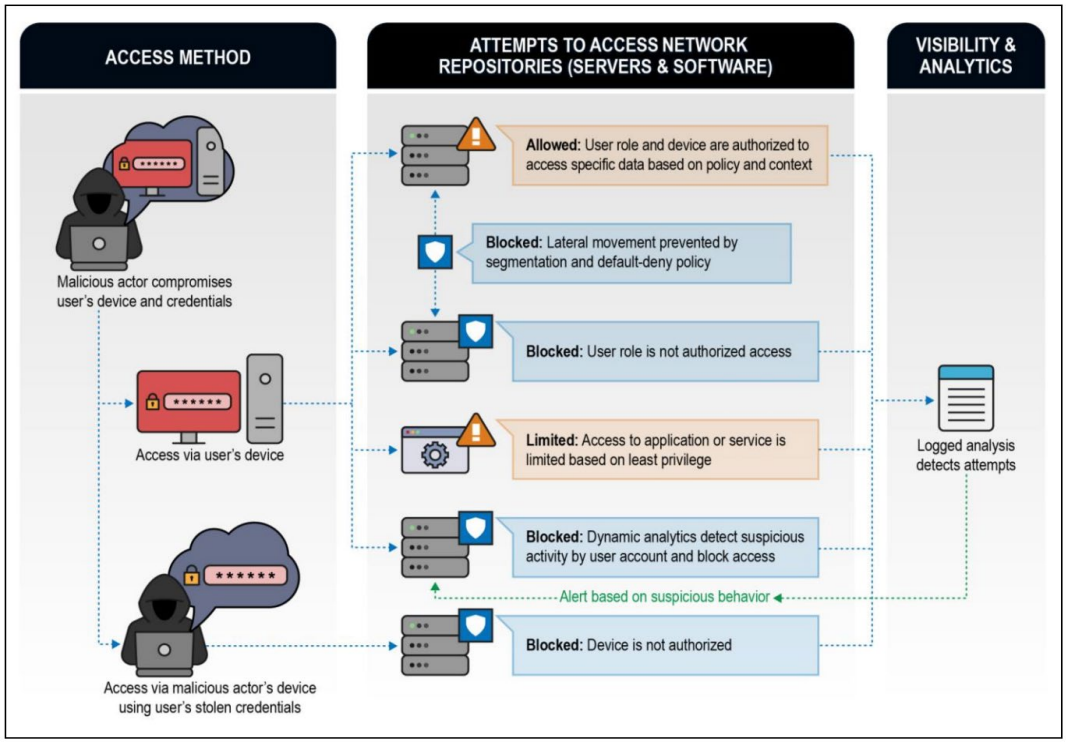

Figure 2: Zero trust architecture use case (NSA, 2021)

From the administrator's point of view, this means that the user uses MFA and the various configurations (device management, data classification & labeling, mobile device management, email security, advanced anti-phishing, impersonation controls, etc.) have been securely implemented. This assumes the administrator has access to security tools and can automate processes to reduce alert fatigue and enhance the security posture under ZTA. The administrator views of identity management, authorization control, certificate protection, and system configurations are in place and functioning as intended to protect sensitive data as it is shared in cross-enterprise systems. Protections include encryption, data classification and labeling, and ex-filtration management and control.

2.2 Concerns

Good ideas, especially high-level ideas, can break down during design and implementation. While ZTA strengths are noteworthy, ZTA is similarly vulnerable to failures during both design and implementation phases. The gaps between vision and implementation can be thought of as *touchpoints* (McGraw, 2006b, p.83). McGraw applied the term touchpoint(s) to software gaps, one of the three pillars of software security; the other two are applied risk management and knowledge (McGraw, 2006a). The same three-pillar philosophy can be applied analogously to ZTA. This philosophy requires an understanding of risk, knowledge of the system & environment, and recognition of the touchpoints. Designing for security, risk analysis/vulnerability testing (Ibid) applies to ZTA as well as software. The security architect must design a ZTA solution that goes beyond controlling unauthorized access to countering the attacker's goals of *deny, deceive, disrupt, destroy,* and *deter*. Some of these attacker goals can be achieved even in ZTA-compliant implementations.

Figure 1 allows the reader to focus on the touchpoints where vulnerabilities could be invoked. A quick view of Figure 1 shows the numerous touchpoints that serve as attack targets. Section 4 of this paper discusses these components in greater detail.

Best practice-defined defense-in-depth solutions offer many of the same advantages of ZTA solutions, such as strong access control, separate enclaves, and proper use of encryption. However, existing solutions often rely on physical separation of systems. The growth of cloud services, software-defined networks (SDN), and software defined containers (SDC) has made such physical and hardware solutions less available. The dynamic nature of SDCs is attractive and frequently mentioned in ZTA solutions. This trade-off is an example of a touchpoint for further examination of layer 2 attacks. The physically separated defense-in-depth solution may suffer layer 2 vulnerabilities but is better positioned to isolate the exploit.

Security architectures are instantiations of overarching security policies designed to reach across the organization (Sherwood, Clark & Lynas, 2005). Such broad policy statements, when applied in the organizational environment where requirements become situational, can be adapted to the single session accesses associated with ZTA. Focusing on specific session details introduces additional touchpoints where interfaces and interactions are not easily standardized. Real value is achieved when the relationship between policy and architecture is carefully considered. Thought leadership must reside amongst architects and policy makers who truly understand real world implications of ZTA. For example, policy makers should have real world experience and be politically neutral, so they may understand how ZTA can be used not just by government agencies, but also by private industry, from small businesses to multinational enterprises. The policy makers should be aware as well of hostile entities who will seek to undermine the security afforded by ZTA.

A remaining factor not depicted, and typically not included, in ZTA discussions is the varying nature of attacker behavioral profiles. The assumption that attackers always seek to gain access, move laterally, then persist in-system, is long-standing (NIST CSRC), reflecting defender biases that do not apply universally. This decades-old characterization of attackers only explains certain hacker behaviors (Sample, et al., 2016). This view contributes to planners failing to imagine alternate attacker goals and behaviors, especially those goals that may be less ambitious but equally effective. A well-timed disruption or denial of service through locking out an important user at a critical moment can be as effective as traditional access breaches.

3. Method

The method used in this study reflects a hypothetical case study. Using the ZTA reference architecture (Department of Defense, 2021) as the test case, this study examines the components and their processing from the adversarial view as a means of identifying potential vulnerabilities. This study presents vulnerability data on ZTA components followed by an exemplar case to step through several examples of activities using multiple architectural views: general user, administrator, and owner. Each case is a hypothetical representation of real-world activities, along with examples of effective intrusions and how they would work in the ZTA example. The selected attacks reflect the attacker goals of deny, deceive, disrupt, destroy, and deter, and are evaluated for efficacy in perimeter, defense-in-depth, and ZTA architectures. Attacks at various layers form the sample.

The hypotheses are:

H_0: ZTA is unbreakable

H_1: ZTA is susceptible to vulnerabilities

4. Findings

Before addressing the components consider the list of security concerns known to impact ZTA and often not sufficiently addressed.

- 1. Signatures vs anomaly detection. Many security products are signature-based, and anomaly detection products have not yet found an effective way to deal with false positives. Byzantine fault tolerance (Veronese et al., 2011) has been used in various domains to create a baseline of acceptable use applicable to ZTA systems.
- 2. Supply chains

- a) Hardware supply chain. When detected, software vulnerabilities can, in many cases, be rebuilt in hours or days. Hardware vulnerabilities, however, may require weeks to months to replace or repair the affected components (Dixon, 2021). Furthermore, nominally identical hardware may be assembled in multiple locations, either in sequential steps or in parallel full production. The absence of public hardware baseline data makes it impossible to verify the purity of a particular piece of hardware. This lack of baselined values makes it possible for attackers to remain undetected as they compromise ZTA component systems (Bhunia et al. 2014).
- Internal network design and firmware of chips used in hardware are generally outside the scope of enterprise security architects designing systems using ZTA principles. Dixon (2021) described a proposed solution being developed by the industry consortium DMTF, the Security Protocol and Data Model Architecture (DMTF, 2022). The driver interface to the hardware must enforce security adherence. As Dixon (2021) suggested, ZTA principles must be designed in before hardware is manufactured.
- b) Software supply chain. Software supply chain vulnerabilities are well documented. A recent identified vulnerability, Log4j (Bing, Satter, & Menn, 2021), is an example showing how ZTA could detect or miss the vulnerability depending on the behaviors involved. Should the vulnerability be invoked, and lateral movement were to follow, the ZTA solution would operate as advertised. However, if the vulnerability were invoked and logging of a specific activity on the server were erased this would make detection more difficult.
- c)Infrastructure. Infrastructure now goes beyond hardware, wires, radios, and traditional infrastructure services such as routing and DNS, to include hypervisors, virtual machines, SDNs and SDCs. A recent compromise of VMWare (Larkshamanan, 2022) listed privilege escalation as one of the effects. When trusted platforms can provision new containers or networks the ability for intruders to collect information indicating which users or hosts are critical becomes easier and less detectable since most security products work above layer 2.
- 3. Zero Day Attacks (0day). 0day attacks are typically undiscovered for 10 months (Greenberg, 2012, Halpern, 2021). In some cases, this time interval is longer. This makes possible intruders hiding themselves by simply implanting without moving and embedding exfiltrated data into “good” payloads.
- 4. Data Centric Attacks. Methods that include data poisoning and other techniques, these attacks take advantage of insufficient data checking against known good or baselined responses.
- 5. Artificial Intelligence/Machine Learning attacks. Data poisoning and model manipulation are well-known methods to subvert AI. Poisoned data can cause AI to generate false negatives while model manipulation can create false or misleading positives.
- 6. Human attacks. Digital deception is becoming increasingly more difficult to detect (Willingham 2022). This suggests that should automated processes fail, the human override can also fail.

5. Vulnerability roadmap

Cyber vulnerability exploitation rarely occurs as an individual event; rather vulnerabilities are exploited in concert as part of a cyber campaign. Campaigns can have kill chains that differ from the cyber kill chain. For this reason, Table 1 lists the ZTA components, their vulnerabilities, and the effects of exploitation of those vulnerabilities.

Table 1: ZTA vulnerability roadmap

ZTA Component and Function	Vulnerabilities	Ramifications
CDM – Detect and mitigate problems. Feeds PE.	Signature based leaves site open to 0-day attacks. Layer 1 & layer 2 attacks	Deny, destroy, deceive.
Industry Compliance – Best practices and standards adherence. Feeds PE	Compliance or best practices not-equal secure. Attackers design campaign around known practices and standards	Disrupt.
Threat Intelligence – Internal and external threat actor feeds including TTPs, indicators of compromise, malware, ransomware. Feeds PE	Signature based leaves site open to 0-day attacks Emergence of new stealth actors False flag operations	Deceive.
Activity Logs – system logs, messages, alarms, notification. Real Time (RT) security posture. Feeds PE.	False flag entries. Attackers remove evidence. Attackers overwhelm logs with entries	Deceive.

ZTA Component and Function	Vulnerabilities	Ramifications
PEP – RT executor of policy Policy Engine – permit or deny access based on CDM, IDAM. Policy Admin - session token creator	PE – Trick via inaccurate feeds. PE - Standard software attacks to breach. PA – gain knowledge of the token creation process to subvert the process.	Deny and deceive.
Data Access Policy – read, write, execute, and delete are granted as least privilege. Users are only included in groups that are needed. Roles are carefully considered.	Privilege escalation via 0-day operating system or application feeds, instead of lateral movement lies in wait. Data centricity results in many groups overlapping, malicious users exploiting transitive trust relationships.	Deny, destroy, and deceive.
PKI - encryption key management	Unknown vulnerabilities in implementation. Intruders gain access to all communications	Disrupt and deceive.
Identity Management Entity credentials, certificates, attributes, roles, etc.; integrates with PKI.	Stolen credentials Lock out key users (e.g., admin) In machine-to-machine communications information gathering (reconnaissance)	Disrupt and deceive.
Security Information & Event Management (SIEM) - Collects security information for analysis and warning.	Misses attack due to obfuscation Ignores attack	Deceive.
Security Orchestration Automated Response (SOAR) - Updates security posture based on SIEM outputs.	Confused response when conflicting data encountered. Poisoned data results in bad decisions Algorithms are manipulated with good data in deceptive weights.	Deny and deceive.

6. Conclusions

ZTA may or may not improve existing security architectures. Traditional physically separated sites may paradoxically increase their risk profile when transitioning to ZTA, while other sites may see an improvement. Architects should have a deep understanding of technology subversion with trade-offs for each instantiation. Hardware and hypervisor vulnerabilities can undermine the carefully crafted separation between data and control planes. Typically undetected for long periods, these attacks are usually missed by security products such as intrusion detection systems and SIEMs, which can only detect indicators of compromise. Similarly, the domains or sandboxes created by SDCs and networks suffer the same fate.

0day attacks undermine the integrity of the components CDM, compliance, activity logs, SIEM, and SOAR, all of which feed the PEP. New threat actors and different behaviours bring unanticipated attacker goals, and previously unseen tactics are missed by of threat intelligence feeds. Encryption may assure privacy and data integrity but once the systems are compromised the same encryption that makes private the communications may also cloak malicious activity.

Incorporating accurate, well-defined baselines into ZTA strengthens the security guidance through an understanding of known good attributes and behaviours. One example is the incorporation of Byzantine fault tolerance methods as mentioned above. Other possible improvements include extending the ZTA principals beyond the traditional software and networking realm into hardware and firmware, where those principles are not regularly addressed, thereby, offering a potentially lasting ZTA benefit.

References

Bell, D. E, and La Padula, L. J. (1976) "Secure computer system: Unified exposition and Multics interpretation." Technical Report ESD-TR-75-306, MITRE Corporation, Bedford, MA. https://csrc.nist.gov/csrc/media/publications/conference-paper/1998/10/08/proceedings-of-the-21st-nissc-1998/documents/early-cs-papers/bell76.pdf [Accessed 14th February 2022]

Bhunia, S., Hsiao, M.S., Banga, M. and Narasimhan, S., 2014. Hardware Trojan attacks: Threat analysis and countermeasures. Proceedings of the IEEE, 102(8), pp.1229-1247

Biden J, (2021) Improving the Nation's Cybersecurity. Executive Order 14028, The White House, https://www.federalregister.gov/documents/2021/05/17/2021-10460/improving-the-nations-cybersecurity [Accessed 14th February 2022]

Bing, C., Satter, R., and Menn, J. (2021) Widely used software with key vulnerability sends cyber defenders scrambling. Reuters. https://www.reuters.com/technology/widely-used-software-with-key-vulnerability-sends-cyber-defenders-scrambling-2021-12-13/ [Accessed 24th February 2022]

Butcher, Z. Zero Trust Architecture, https://www.tetrate.io/white-paper-zero-trust-architecture/ [Accessed 14th February 2022]

Campbell, M. (2020) "Beyond zero trust: trust is a vulnerability. *Computer*, 53(10), pp.110-113.the Department of Defense (DOD) Zero Trust Reference Architecture, V. 1.0 (2021) https://dodcio.defense.gov/Portals/0/Documents/Library/(U)ZT_RA_v1.1(U)_Mar21.pdf [Accessed 14th February 2022]

Dixon, M.G., (July 21, 2021). "A zero trust approach to architecting silicon". https://www.intel.com/content/www/us/en/newsroom/opinion/zero-trust-approach-architecting-silicon.html#gs.q7qp87 [Accessed 24th February 2022]

DMTF (2022) DMTF Releases Security Protocol and Data Model (SPDM) Architecture as Work in Progress. https://www.dmtf.org/content/dmtf-releases-security-protocol-and-data-model-spdm-architecture-work-progress [Accessed 24th February 2022]

Greenberg, A., (June 16, 2012). "Hackers exploit software bugs for 10 months on average before they are fixed", *Forbes*. https://www.forbes.com/sites/andygreenberg/2012/10/16/hackers-exploit-software-bugs-for-10-months-on-average-before-theyre-fixed/?sh=72a6d2daee1a [Accessed 22d February 2022]

Halpern, S., (January 25, 2021). "After the solarwinds hack, we have no idea what cyber dangers we face", The New Yorker. Website https://www.newyorker.com/news/daily-comment/after-the-solarwinds-hack-we-have-no-idea-what-cyber-dangers-we-face

Kindervag, J., Balaouras, S., and Colt, L. (2010) "No more chewy centers: Introducing the zero trust model of information security", Forrester Research, https://media.paloaltonetworks.com/documents/Forrester-No-More-Chewy-Centers.pdf [Accessed 14th February 2022]

Larkshamanan, R. (February 16, 2022). "VMWare issues security patches for high-severity flaws affecting multiple products", The Hacker News. Website: https://thehackernews.com/2022/02/vmware-issues-security-patches-for-high.html

Lefkovitz, N. and Boeckl, K. (2020) "NIST Privacy Framework: An Overview" NIST ITL Bulletin https://csrc.nist.gov/CSRC/media/Publications/Shared/documents/itl-bulletin/itlbul2020-06.pdf [Accessed 14th February 2022]

Lockheed Martin (2015), Gaining the Advantage: Applying Cyber Kill Chain® Methodology to Network Defense, https://www.lockheedmartin.com/content/dam/lockheed-martin/rms/documents/cyber/Gaining_the_Advantage_Cyber_Kill_Chain.pdf [Accessed 14th February 2022]

McGraw, G. (2006a) Three Pillars of Software Security. http://www.swsec.com/resources/pillars/ [Accessed 14th February 2022]

McGraw, G. (2006b) Software Security: Building Security In. Addison Wesley.

NSA – National Security Agency (2021), Embracing a Zero Trust Security Model, v. 1.0, https://media.defense.gov/2021/Feb/25/2002588479/-1/-1/0/CSI_EMBRACING_ZT_SECURITY_MODEL_UOO115131-21.PDF [Accessed 14th February 2022]

Rose, S., Borchert, O. , Mitchell, S. and Connelly, S. (2020) Zero Trust Architecture, Special Publication (NIST SP) 800-207, National Institute of Standards and Technology, Gaithersburg, MD, [online], https://doi.org/10.6028/NIST.SP.800-207 , [Accessed January 3d, 2022]

Rose, S. (2019) Zero Trust 101, https://csrc.nist.gov/CSRC/media/Presentations/zero-trust-architecture-101/images-media/Zero%20Trust%20Architecture%20101%20-%20Scott.pdf [Accessed 14 February 2022]

Ross, R., et al. (2018) Risk Management Framework for Information Systems and Organizations: A System Life Cycle Approach for Security and Privacy, NIST SP 800-27 Rev. 2. https://doi.org/10.6028/NIST.SP.800-37r2 [Accessed 14th February 2022]

Sample, C., Cowley, J., Watson, T., and Maple, C., (2016) "Re-thinking Threat Intelligence". In 2016 International Conference on Cyber Conflict (CyCon US) (pp. 1-9). IEEE.

Sherwood, J., Clark, A., and Lynas, D. (2005) Enterprise Security Architecture: A Business-Driven Approach, CMP Books, San Francisco

Simmonds, P. (2019). The Fallacy of the "Zero Trust Network. RSA Conference 2019. https://youtu.be/tFrbt9s4Fns [Accessed Feb 8, 2021]

Veronese, G.S., Correia, M., Bessani, A.N., Lung, L.C. and Verissimo, P., 2011. Efficient Byzantine fault-tolerance. IEEE Transactions on Computers, 62(1), pp.16-30.

Weichert, M. (2019) Update to the Trusted Internet Connections (TIC) Initiative, OMB Memorandum M-19-26, https://www.whitehouse.gov/wp-content/uploads/2019/09/M-19-26.pdf [Accessed 14th February 2022]