International Journal of Emerging Research in Engineering and Technology
Pearl Blue Research Group| Volume 6 Issue 3 PP 113-122, 2025
ISSN: 3050-922X | https://doi.org/10.63282/3050-922X.IJERET-V6I3P113

Original Article

A Zero Trust-Based Approach to Modern Cybersecurity Challenges in Software Development

Pradyumna Kumar
Independent Researcher.

Received On: 05/07/2025 ***Revised On: 24/07/2025*** ***Accepted On: 26/08/2025*** ***Published On: 07/09/2025***

Abstract *- The rapidly shifting digital landscape, with increased complexity of software systems and the utilization of cloud systems, mobile technologies, and continuous deployment methods, has presented cybersecurity threats as never before. The application of conventional models, which can be characterized as the perimeter-based castle-and-moat strategy is no longer enough to counter sophisticated threats like supply chain vulnerabilities, ransom are, and insider assaults. As a paradigm, the Zero Trust Network Architecture (ZTNA) is designed to address these weaknesses as a groundbreaking solution based on "never trust, always verify." This article discusses applying the concept of Zero Trust (ZT) to modern software development, exploring how the concept can be integrated into the DevSecOps pipeline to ensure identity, device, and code integrity checks are performed at each stage of development. Some of the most significant aspects considered in relation to enabling the creation of resilient, safe systems include Identity and Access Management (IAM), endpoint security, network segmentation, SIEM, DLP, CASB, and threat intelligence. Furthermore, the paper addresses the topic of cybersecurity vulnerabilities in DevOps and CI/CD pipelines, as well as the concepts of data privacy and regulatory compliance standards, including HIPAA and GDPR. Finally, the challenges of the implementation of the ZT, which are the integration with the legacy systems and the computational requirements, are addressed with the assistance of strategic advice on safe and scaled implementation.*

Keywords *- Cybersecurity, Zero Trust Network Architecture (ZTNA), Secure Software Development, Identity and Access Management (IAM), HIPAA, Cloud Security, Regulatory Compliance.*

1. Introduction

The implementation of Mobile devices, cloud services, and remote work in the evolving nature of cybersecurity creates complexities that break the paradigm of network security [1]. The traditional models of cybersecurity are premised on the concept of a secure inner network with a well-defined perimeter that can be trusted. This strategy, often known as the castle-and-moat concept, is predicated on the idea that there are dangers outside the walls, and the organizations within are automatically trusted [2]. This has, however, been rendered ineffective by ransomware, insider attacks, and advanced persistent threats (APTs), among other more complex forms of cyberthreats. A paradigm shift has been brought about by Zero Trust Network Architecture (ZTNA) to overcome such issues. Generally speaking, ZTNA adheres to the maxim "never trust, always verify." When it comes to both within and outside of a network perimeter, ZTNA assumes by default that no individual, gadget, or program is reliable, in contrast to previous methods [3]. Rather, it implements unceasing authentication, authorization and validation of security stance to each request of access [4][5], without taking into account the requester's whereabouts or the resource requested.

ZTNA is an architectural construct made up of a number of security practices and technologies rather than a single technology. These include endpoint detection and response (EDR), micro segmentation, multi-factor authentication (MFA), identity and access management (IAM), and secure access service edge (SASE) [6]. All these factors create a strong security stance that can provide control over the movement of laterals, minimize attack surfaces and improve network interaction visibility and management. The application of Zero Trust (ZT) concepts apply to the whole software lifecycle, including network security, as implemented in the context of software development [7].

Verifying identities should be integrated into the secure development process encompassing the design and code integrity that should be exercised across the whole design, development, testing process, deployment and maintenance. The addition of ZT in DevSecOps pipelines ensure that all internal and third-party components are authenticated, authorized and monitored to ensure that it is less prone to injecting weaknesses into applications. By integrating those principles into the software development cycles, organizations are able to develop applications that are resilient to the modern cyber threats, reduce attack surfaces and preserve a strong security posture even in dynamic, cloud-centric and distributed environments.

1.1. Structure of the Paper

This study is organized into six sections: Section II covers Cybersecurity in Modern Software Development, Section III presents ZT principles, Section IV discusses adoption challenges with recommendations, Section V reviews relevant literature, and Section VI concludes with future directions for scalable, AI-driven, and compliant ZT solutions.

2. Cybersecurity in Modern Software Development

The complexity and interconnectedness of software systems in today's digital environment make them easy targets for cyberattacks. As organizations strive for faster development cycles and continuous deployment, security often struggles to keep pace with innovation [8]. Modern applications face threats from insecure coding practices, third-party library vulnerabilities, supply chain vulnerabilities, and sophisticated attack vectors, including ransomware and zero-day exploits. There are also additional complications related to maintaining data confidentiality and regulatory sensitivity when handling sensitive information requires careful attention. The only way of reducing these risks is to incorporate security in the software development life cycle, automated testing, continuous monitoring, IAM and training the developers to develop resilient and trustworthy systems.

2.1. Emerging Threats and Vulnerabilities

The development of the practice of software development which is now accelerating with the innovations of agile and DevOps, the threat of cyber-attacks on the development environments is growing exponentially. Before applications get to production, attackers use these vulnerabilities in code, development tools and deployment pipelines to compromise applications [9]. The most notable of these threats are malware programs, including viruses, worms, Trojans, and rootkits, which may infect development machines or code repositories, interrupt workflows, or introduce malicious code. Ransomware attacks have the capability of locking sources of code or CI/CD Pipeline assets, whereas phishing and social engineering methods direct their tools against developers to obtain credentials to repositories and cloud systems. Another urgent issue is supplying chain attacks, where the integrity of applications has been compromised by malicious code in third-party libraries or dependencies. Other vectors of attack are the hijacking of development settings, code transmissions, man-in-the-middle attacks and exploits that involve passwords or credentials. To mitigate such threats, constant monitoring, vulnerability scanning, effective coding practices, and developer awareness are crucial to ensuring that software development processes are resilient and secure.

Fig 1: Computer Threats

These cyberthreats include ransomware, phishing, malware, and actual assaults on the vulnerabilities of computer systems. Vigilance is important, and security measures should be used to avoid such threats (Figure 1).

2.2. Security Risks in DevOps and CI/CD Pipelines

CI/CD security uses stringent security measures to protect a development pipeline's availability, integrity, and secrecy [10][11]. This includes administering secrets (such API keys and credentials) during the build and deployment procedures, doing security testing, and safeguarding code repositories. Incorporating security checks straight into the pipeline may identify vulnerabilities early on and stop them from spreading to production. Instead of seeing security as a discrete stage, CI/CD security views it as a continuous component of development, security, and operations (DevSecOps) procedures. This method helps defend against any breaches and preserve user confidence by safeguarding every code update and deployment process.

2.2.1. Common CI/CD Security Risks

CI/CD pipelines speed up DevOps, but they can create security flaws. The main hazards to be aware of are as follows:

- **Insecure Code Practices:** Prior to deployment, one of the main purposes of a CI/CD pipeline is to find code vulnerabilities [12]. But unsafe code can get past without regular security checks, leaving apps vulnerable to possible attack.
- **Insufficient Access Controls:** Sensitive data must be accessible for CI/CD processes to work properly. Overly restrictive access restrictions provide unauthorized actors the opportunity to enter, change code, or access private information.
- **Security Misconfigurations:** CI/CD environments are complex, with many interrelated systems. This implies that there are more chances for errors to occur, whether in deployment settings or CI/CD systems [13][14]. Open ports, insecure permissions, and unsafe defaults are common vulnerabilities that attackers can exploit to undermine pipeline security.
- **Exposed Secrets:** Secrets like certificates, API keys, and passwords are frequently needed by pipelines. These might make it possible for hackers to intercept them and obtain unauthorized access to vital systems if they are stored insecurely or are left in plain text inside the pipeline.
- **Vulnerable Third-Party Dependencies:** The majority of contemporary apps depend on third-party libraries, which may cause CI/CD process vulnerabilities. The security of the entire program may be jeopardised if one of these dependents has a bug or backdoor.
- **Supply Chain Attacks:** In supply chain attacks, attackers target the open-source libraries and dependencies that programs use [15]. They can take advantage of any program that incorporates these requirements by adding malicious code or vulnerabilities.

2.2.2. CI/CD Pipeline Security Best Practices

This is how to manage every security area to avoid the dangers of getting into the pipeline.

- **Enforce Strict Access Controls:** Enforce role-based access control (RBAC) and least privilege in order to have access to only what is required [16][17]. Multi-factor authentication (MFA) also implies the application of extra security measures, and access to the sensitive areas can be checked with the help of regular audits.
- **Automate Code Scanning:** Include automated code scanners in the CI/CD pipeline, like dynamic and static application security testing (DAST and SAST). This can ensure that it is much easier to discover the weaknesses and problems before they are production-ready. The unsecured code is then published and requires being fixed, which can be very costly.
- **Manage Secrets Securely:** Secrets that must be treated properly include encryption keys and APIs. Instead of being saved in computer code, these credentials ought to be encrypted and kept in a secrets management system. This method safeguards private information and only utilizes it when required.
- **Monitor Third-Party Dependencies:** Prior to third-party components becoming harmful, identify their shortcomings. Utilize software scanning techniques to ensure code security by examining dependencies.
- **Update CI/CD Tools and Dependencies:** Attackers have access to outdated CI/CD security technologies. Regularly update and patch all the pipeline elements in order to prevent the exploitation of known vulnerabilities [18][19]. This easy yet crucial step significantly reduces the likelihood of attacks against outdated software.
- **Enable Continuous Monitoring and Logging:** Constant monitoring gives staff insight into the pipeline, enabling them to promptly address any questionable behavior and stop unwanted access.
- **Secure Configuration Settings:** It's simple to forget about configuration options, yet unsafe setups expose users unintentionally. Adhere to best practices by limiting public access to critical locations, implementing network segmentation, and turning off unwanted services.
- **Conduct Regular Security Audits:** Regularly carrying out penetration testing and security audits is beneficial to ensure that security procedures are current and efficient while providing a clear picture of any flaws in the CI/CD pipeline.
- **Build a Culture of DevSecOps Collaboration:** DevOps and security teams should collaborate to integrate security into every step of the development process. Make communication a top priority and provide training on secure coding techniques to all team members. People are better able to promote a proactive approach to CI/CD security.

2.3. Data Privacy and Regulatory Compliance

The following discusses HIPAA and GDPR compliance:

2.4. HIPAA Compliance

The foundational rules for protecting sensitive patient data were created by the Health Insurance Portability and Accountability Act (HIPAA). This regulation is applicable to people or organizations that get health information while engaging in routine medical procedures [20][21]. Healthcare providers, such as radiology centers, hospitals, and health plans, are among the covered entities. Organizations that offer medical treatment or at least cover its cost, like insurers, are known as health plans. This regulation safeguards all of a patient's personally identifying information. Included in this data are the patient's demographics, medical history, and more. This rule is not applicable if the data is deidentified in accordance with the regulation.

2.4.1. GDPR Compliance

The General Data Protection Regulation (GDPR), which came into effect on May 25, 2018, has been enforced by the European Union and established a comprehensive framework for safeguarding the personal data of EU residents and citizens. It contains any information that might be used to identify a specific person, including names, identification numbers, and location data, and it is applicable to all organizations handling such data, regardless of location [22][23]. GDPR gives people the ability to view personal data, limit how it is processed, and ask for its erasure (sometimes known as the "right to be forgotten"). Organizations are responsible for secure data handling and must report any breaches to the relevant authority within 72 hours [24]. Significant fines of up to €10–20 million or 2-4% of yearly revenue may be imposed for noncompliance. The following provides a comparison between the two compliance frameworks in Table I:

Table 1: The Major Differences between HIPAA and GDPR

Aspect	HIPAA (Health Insurance Portability and Accountability Act)	GDPR (General Data Protection Regulation)
Data Sharing Without Consent	Organisations may, under some conditions, divulge patient information to another provider without authorisation.	No personal information may leave the organization's property without the resident or EU citizen's permission.
Right to Erasure	Does not provide a right for patients to request data deletion.	EU citizens or residents can request their data to be erased under certain circumstances.

Data Breach Notification	Providers must notify affected subjects; if >500 subjects are affected, the Department of Health & Human Services must be informed.	Data breaches must be reported to a supervisory authority within 72 hours.
Consent Exceptions	Disclosure allowed if an individual is incapable of giving consent due to incapacity.	Similar provision: processing allowed if individual cannot consent due to incapability.
Processing for Not-for-Profit Organizations	No specific restriction on processing for not-for-profit organizations.	Processing is only allowed when it pertains to the person's family or personal affairs and not to outside parties.
Legal/ Court Disclosure	Data can be disclosed when required by a court in its judicial capacity.	Same as HIPAA: disclosure allowed when needed in legal or judicial proceedings.

3. Zero Trust Approach for Secure Software Development

The fundamental elements of IAM, network segmentation (or micro-segmentation), in the context of safe software development, a Zero Trust Architecture (ZTA) consists of the following elements: continuous monitoring and verification, data protection, device and endpoint security, and the "never trust, always verify" tenet. Within the software development lifecycle, these elements ensure that only authenticated developers and processes can access code repositories, that development and testing environments remain isolated, and that sensitive data used in applications is safeguarded throughout the pipeline [25][26]. An enterprise-level ZTA implementation for software development comprises several logical components that can be deployed in the cloud or on-premises, providing flexibility while maintaining strict security controls across the development, integration, and deployment stages.

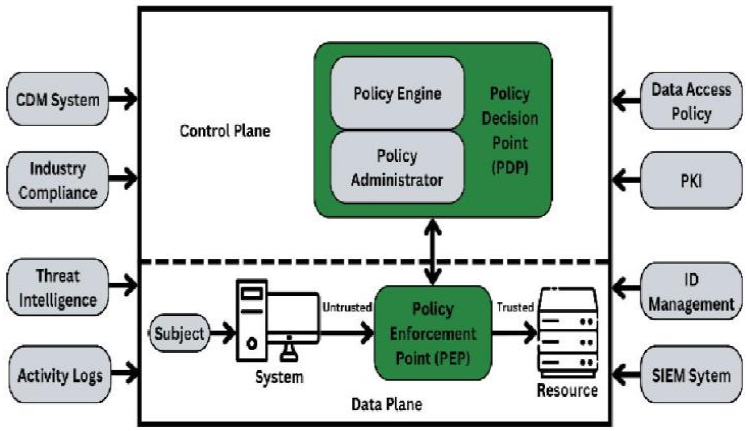

Fig 2: Zero Trust Core Logical Components

Figure 2 shows the two wings, or planes, that make up the rational framework of a paradigm for zero-trust security. Central to the process are the inputs from many sources, including CDM systems, which are processed by the Policy Engine, Policy Administrator, compliance standards, threat intelligence, and logs are all part of the Policy Decision Point (PDP), which is where access decisions are made. In the Data Plane, it'll find the Policy Enforcement Point (PEP), which guards against people or systems gaining unauthorized access to the target resource. The concept embodies ZT guiding principle of "never trust, always verify" by enforcing access based on identity verification, dynamic risk assessments, and stringent data access regulations.

3.1. Identity and Access Management (IAM)

As demonstrated in Figure 3, A crucial component of ZTNA, Identity and Access Management (IAM) enables companies to control and manage who has access to their resources based on verified identities [27]. IAM uses strong authentication and authorization procedures to ensure that only authorized individuals, Applications, or devices are granted access to specific information or services [28]. This often means utilizing multi-factor authentication (MFA) to provide an additional layer of security and single sign-on (SSO) to expedite user access without compromising security.

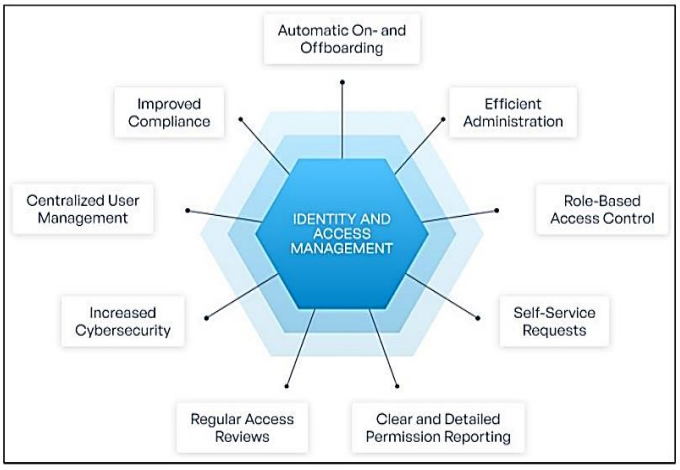

Fig 3: Identity and Access Management

The IAM systems evaluate access requests in real time while accounting for additional contextual elements like as location, device health, user behavior, and organizational roles. By providing users with the bare minimum of rights necessary to carry out their duties, IAM systems use the concept of least privilege through role-based access control (RBAC) or attribute-based access control (ABAC). This operational and dynamic IAM approach is better than the concept of ZT by explicitly validating all access and minimizing the chances of compromised access, insider threats and misuse of credentials.

- **Endpoint security:** The endpoint security is all about ensuring that all the devices whether it is workstations, mobile phones or IoT devices are cleared of security checks, before they can get access to network resources [29].
- **Network segmentation:** The most important security strategy, network segmentation, divides the network into several sections, each with a unique set of access restrictions and security guidelines.
- **Security Information and Event Management (SIEM):** Monitoring and analyzing security occurrences are made easier with the help of Security Information and Event Management (SIEM), as well as the response to occurrences in the entire network in real time.
- **Data Loss Prevention (DLP):** Data loss prevention is an indispensable part of ZTNA as its purpose is to ensure that confidential data does not leak, steal, or otherwise get inaccessible without a legitimate reason [30].
- Zero Trust Policy Engine: The ZT policy engine is in charge of creating, implementing, and regularly assessing the network's security policy.
- **Continuous Diagnostics and Mitigation (CDM):** A proactive and dynamic security measure in ZTNA is Continuous Diagnostics and Mitigation (CDM). metric that involves identifying threats, evaluating current vulnerabilities, and reacting to them immediately.
- **Cloud Access Security Broker (CASB):** A Cloud Access Security Broker is an essential part of the network's zero-trust architecture (CASB), which offers monitoring and control over how cloud services are used [31], which guarantees safe access to data and apps stored in the cloud.
- **Encryption and Secure Communication:** Encryption and secure communication are essential parts of a ZTNA and therefore guarantee data protection in transit and at rest, both of which do not depend on where the data is stored or by the party gaining access to the data [32].
- **Threat Intelligence and Behavioral Analytics:** The ability to detect and address potential threats before they can cause harm is one of ZTNA's primary components, along with threat intelligence and behavioral analytics [33].

4. Challenges and Recommendation for Zero Trust in Software Development

The Zero Trust principles in software development provide a variety of difficulties. The user and device authentication must be carried out on a continuous basis, and it is also necessary to make sure that the user experience is not compromised, which can only be achieved through sophisticated authentication and access control services. ZT may be difficult to integrate with the legacy systems and applications that an organization has implemented, potentially involving a great deal of architectural work and planning. Also, to trace every action within the network in real time to monitor, record and analyze it to identify irregularities in the normal operation might be exemplary in terms of computational and security capabilities [34]. Organizations can solve these issues through a staged method that begins with key assets and risky apps, uses powerful identity and access control solutions, takes advantage of automation to monitor constantly, and regularly revises policies to cope with the dynamic threat. The education and training of teams on the principles of the ZT is also essential so that the implementation of the new concept is consistent and aligned with the culture of security-first practices. The next section of this article outlines the primary challenges and suggested ways to get around them:

4.1. Integration with Legacy Systems

The application of ZT is seen to be incompatible with the old infrastructure and applications as the implementation compels architecture changes and is expensive.

Recommendation: Implement incremental migration strategy, focus on important assets and apply API gateways or micro-segmentation to connect old systems with the new security systems.

4.2. User Experience vs. Continuous Verification

Insufficient design of regular authentication verifications might irritate users and reduce productivity.

Recommendation: Use dynamic and risk-based authentication and use Single Sign-On (SSO) and multi-factor authentication (MFA) to trade usability and security.

4.3. High Computational and Monitoring Demands

Continuous logging, anomaly detection, and real-time monitoring place heavy burdens on IT resources [34].

Recommendation: Employ automation, AI-driven analytics, and scalable cloud-native tools to optimize monitoring without overloading infrastructure.

4.4. Policy Complexity and Scalability

Managing granular access policies across large, dynamic environments can become unmanageable.

Recommendation: Standardize policies using centralized policy engines, automate enforcement, and regularly review and update policies as systems evolve.

4.5. Cultural and Skill Gaps

Teams may lack awareness or training in ZT principles, slowing adoption and creating resistance.

Recommendation: Conduct regular training, establish a security-first culture, and integrate DevSecOps practices to align development and operations teams with ZT goals.

4.6. Cost and Resource Constraints

Investing in new tools is necessary to implement ZT, identity systems, and monitoring platforms, which may strain budgets.

Recommendation: Focus on risk-based prioritization, begin with high-value assets, and gradually expand ZT capabilities to achieve cost-effective adoption.

5. Literature of Review

The literature highlights opportunities and challenges of ZT in software development, focusing on access control, authentication, and secure DevSecOps practices, while proposing frameworks and best practices to enhance cybersecurity. Gambo and Almulhem (2025) analyzed ten years of ZTA research (2016–2025) using the PRISMA framework, providing an SLR that provides a synopsis of the technology, its uses, and the associated challenges of ZTA. It critically analyses the obstacles to ZTA adoption and offers a thorough taxonomy that arranges the application areas of ZTA together with the cutting-edge technologies that make its implementation easier [35].

Park, Park and Youm (2025) suggested an improved security paradigm that combines the ideas of ZT and Multi-Level Security (MLS). "Classified," "Sensitive," and "Open" are the three sensitivity categories into which the suggested model divides data. It enhances data security and usability by applying dynamic restrictions and tailoring security to meet customized needs at every level. Additionally, by integrating ZT's automated dynamic access capabilities, the model gets beyond MLS's static access control constraints and greatly increases response to unusual behaviors. The study supports the creation and testing of a special security design that ensures safe data use and protection even when remote networks are involved, such as those seen in governmental and military institutions. Furthermore, it offers a starting point for the creation of advanced security systems in the future [36].

Dhiman et al. (2024) improves awareness of ZT, which in turn promotes Secure network solutions' expansion and application in vital infrastructures. The survey report offers a thorough explanation of ZT fundamental concepts in addition to assessing the various solutions and their possible applications. The essay first explores the role of authentication and access control in ZT systems before looking at more innovative approaches to these practices in different contexts. The topic of security automation, micro-segmentation, and traditional encryption techniques is covered in greater detail, along with their applicability in a safe ZT environment [37].

Table 2: Comparative analysis of zero trust implementation in Software Development

Author	Study On	Approach	Key Findings	Challenges	Future Directions
Gambo & Almulhem (2025)	Zero Trust Architecture (ZTA)	Systematic Literature Review (PRISMA framework, 2016–2025)	presented a thorough taxonomy of application areas and technologies and synthesized ZTA applications, supporting technologies, and difficulties	Barriers to ZTA adoption across domains	Facilitate adoption of ZTA in diverse sectors and guide future research on emerging technologies
Park, Park & Youm (2025)	Enhanced Security Model integrating Multi-Level Security (MLS) & ZT	Model-based study classifying data sensitivity levels	Dynamic access control based on data sensitivity; improved responsiveness to anomalous behavior; enhanced security and usability	Limitations of MLS static access control; complexity of integrating ZT dynamic features	Development of advanced security frameworks for isolated networks (e.g., military, government)
Dhiman et al. (2024)	Secure Network Architectures in Critical Infrastructures	Extensive survey	Investigated security automation, micro-segmentation, encryption, access control, and authentication for ZTA.	Challenges in implementing ZT across diverse scenarios	Provide practical guidelines for secure ZT implementation in critical infrastructures
Lund et al. (2024)	Zero Trust Cybersecurity Framework	Conceptual overview	Constant authentication, least privilege access, and the maxim "never trust, always verify" were stressed, breach assumption; focused on large information-exchange environments	Implementation complexity in large-scale environments such as schools and libraries	Expand applicability of ZTA principles to various organizational environments
Yeoh et al. (2023)	ZT Implementation Critical Success Factors (CSFs)	Delphi research involving 12 cybersecurity specialists in three rounds	Developed multi-dimensional CSFs framework (identification, network, infrastructure, visibility, automation, endpoint, application, and data); maturity assessment framework	Organizational challenges in adopting ZTA across multiple dimensions	Guide organizations in deploying ZT practically and evaluating maturity
Patel (2023)	Security Architecture in Agile & DevSecOps	Applied ZT principles (least privilege, ongoing authentication, dynamic policies) in cloud-native environments	Addressed microservices, containers, and Kubernetes orchestration security; provided recommended solutions for DevSecOps	Security challenges in cloud-native and containerized environments	Enhance ZT adoption in agile development lifecycles and cloud-native applications

Table II summarizes key studies on ZT implementation in software development, outlining research focus, approaches, key findings, challenges, and proposed future directions, thereby highlighting advancements and persistent gaps in securing software development environments.

Lund et al. (2024) describe the zero-trust cybersecurity paradigm, which is the notion that vulnerabilities in organizations are reduced due to the adage "never trust, always verify." The paper's special focus on how any organisation, such as a school or library, can utilise zero-trust concepts to facilitate secure information exchange. It proposes the importance of least privilege access, which specifically defies presumptions and allows users access to only what they need, and continuous authentication, which establishes who users are on the network, which presupposes the breach and acts to prevent its further spread by implementing several checkpoints across the network [38].

Yeoh et al. (2023) published three Delphi study rounds to get a panel of twelve cybersecurity experts to agree on adopting ZT cybersecurity to a multifaceted framework of the CSFs with eight components data, network, infrastructure, identity, endpoint, automation and orchestration, visibility and analytics, workload, and application. A maturity assessment approach was developed to help organizations determine their zero-trust maturity using the CSFs. This research advances a theoretical knowledge of zero-trust deployment from a variety of angles and provides organizations with a workable framework for practical assistance [39].

Patel (2023) proposed a study that utilizes an enhanced security architecture, incorporating the fundamental security tenets of constant authentication and access to ZT least privilege, as well as dynamic policy enforcement, across the DevSecOps pipeline within current agile development lifecycles. The analysis identifies security challenges of cloud-native environments that stem from microservices and containers, as well as orchestration systems using Kubernetes, while providing recommended solutions for all development periods [40].

6. Conclusion and Future Work

The security of software systems has become a major problem in today's dynamic digital environment as a result of organizations dealing with sophisticated cyber threats, dispersed infrastructures, and more complex applications. The protection against contemporary threats like as supply chain breaches, ransomware, and insider assaults is no longer possible with traditional perimeter-based security solutions. This study highlighted Zero Trust Network Architecture's (ZTNA) importance as a ground-breaking idea for protecting software development processes. ZT is a technique to put the principle of never trusting into reality by constantly confirming the procedures of rigorous access control, continuous authentication, endpoint validation, and active monitoring during the whole development cycle. The inclusion of ZT into DevSecOps pipelines contributes to the resilience of reducing attack surfaces, decreasing the succession of lateral movement, and protecting sensitive data in the cloud and hybrid-based ecosystems. Although the ZT has a considerable list of advantages, its implementation has a range of issues, such as compatibility with older systems, seamless user experience, and computational load. Research in the future must be aimed at creating lightweight and scalable ZT frameworks applied to cloud-native and hybrid applications. New trends are to exploit AI-powered anomaly detection, automation of compliance with regulations, including HIPAA and GDPR, and to consider some more advanced technologies, such as blockchain, to ensure a safe audit trail and quantum-safe cryptography to protect data over the long term.

References

[1] S. A. Daniel and S. S. Victor, “Emerging Trends in Cybersecurity for Critical Infrastructure Protection: A Comprehensive Review,” *Comput. Sci. IT Res. J.*, vol. 5, no. 3, pp. 576–593, Mar. 2024, doi: 10.51594/csitrj.v5i3.872.

[2] C. Daah, A. Qureshi, I. Awan, and S. Konur, “Enhancing Zero Trust Models in the Financial Industry through Blockchain Integration: A Proposed Framework,” *Electronics*, vol. 13, no. 5, p. 865, Feb. 2024, doi: 10.3390/electronics13050865.

[3] S. Sarkar, G. Choudhary, S. K. Shandilya, A. Hussain, and H. Kim, “Security of Zero Trust Networks in Cloud Computing: A Comparative Review,” *Sustainability*, vol. 14, no. 18, p. 11213, Sep. 2022, doi: 10.3390/su141811213.

[4] F. A. Qazi, “Study of Zero Trust Architecture for Applications and Network Security,” in *IEEE 19th International Conference on Smart Communities: Improving Quality of Life Using ICT, IoT and AI, HONET 2022*, 2022. doi: 10.1109/HONET56683.2022.10019186.

[5] Vikas Prajapati, “Role of Identity and Access Management in Zero Trust Architecture for Cloud Security: Challenges and Solutions,” *Int. J. Adv. Res. Sci. Commun. Technol.*, vol. 5, no. 3, pp. 6–18, Mar. 2025, doi: 10.48175/IJARSCT-23902.

[6] J. Jagannath, K. Ramezanpour, and A. Jagannath, “Digital Twin Virtualization with Machine Learning for IoT and Beyond 5G Networks: Research Directions for Security and Optimal Control,” in *WiseML 2022 - Proceedings of the 2022 ACM Workshop on Wireless Security and Machine Learning*, 2022. doi: 10.1145/3522783.3529519.

[7] D. Patel and R. Tandon, “Cryptographic Trust Models and Zero-Knowledge Proofs for Secure Cloud Access Control and Authentication,” *Int. J. Adv. Res. Sci. Commun. Technol.*, pp. 749–758, Dec. 2022, doi: 10.48175/IJARSCT-7744D.

[8] H. Kang, G. Liu, Q. Wang, L. Meng, and J. Liu, “Theory and Application of Zero Trust Security: A Brief Survey,” *Entropy*, vol. 25, no. 12, p. 1595, Nov. 2023, doi: 10.3390/e25121595.

[9] S. Narang and A. Gogineni, “Zero-Trust Security in Intrusion Detection Networks: An AI-Powered Threat Detection in Cloud Environment,” *Int. J. Sci. Res. Mod. Technol.*, vol. 4, no. 5, pp. 60–70, Jun. 2025, doi: 10.38124/ijsrmt.v4i5.542.

[10] K. Denzel, “A survey of security in zero trust network architectures,” *GSC Adv. Res. Rev.*, vol. 22, no. 2, pp. 182–214, Feb. 2025, doi: 10.30574/gscarr.2025.22.2.0036.

[11] R. Patel, “Automated Threat Detection and Risk Mitigation for ICS (Industrial Control Systems) Employing Deep Learning in Cybersecurity Defence,” *Int. J. Curr. Eng. Technol.*, vol. 13, no. 06, pp. 584–591, Dec. 2023, doi: 10.14741/ijcet/v.13.6.11.

[12] A. Levine and B. A. Tucker, “Zero Trust Architecture: Risk Discussion,” *Digital Threats: Research and Practice*. 2023. doi: 10.1145/3573892.

[13] S. Ashfaq, S. A. Patil, S. Borde, P. Chandre, P. M. Shafi,

and A. Jadhav, "Zero Trust Security Paradigm: A Comprehensive Survey and Research Analysis," *J. Electr. Syst.*, 2023, doi: 10.52783/jes.688.

[14] A. R. Bilipelli, "AI-Driven Intrusion Detection Systems for LargeScale Cybersecurity Networks Data Analysis: A Comparative Study," *TIJER*, vol. 11, no. 12, pp. 1–7, 2024.

[15] K. Wannere, "Exploring the Implementation and Challenges of Zero Trust Security Models in Modern Network Environments," *Int. J. Eng. Res. Technol.*, vol. 14, no. 05, 2025.

[16] S. Pawar, S. Vaz, Y. Khandagale, and M. Pokharkar, "Zero Trust Architecture: A Paradigm Shift in Cybersecurity," *Int. J. Res. Publ. Rev.*, no. 5, pp. 6454–6460, 2024.

[17] A. Goyal, "Optimising Cloud-Based CI/CD Pipelines: Techniques for Rapid Software Deployment," *Tech. Int. J. Eng. Res.*, vol. 11, no. 11, pp. 896–904, 2024.

[18] C. Buck, C. Olenberger, A. Schweizer, F. Völter, and T. Eymann, "Never trust, always verify: A multivocal literature review on current knowledge and research gaps of zero-trust," *Comput. Secur.*, 2021, doi: 10.1016/j.cose.2021.102436.

[19] A. Goyal, "Optimising Software Lifecycle Management through Predictive Maintenance : Insights and Best Practices," *Int. J. Sci. Res. Arch.*, vol. 07, no. 02, pp. 693–702, 2022.

[20] X. Wang, F. Xie, P. Gu, D. Shi, and K. Gu, "Evaluating privacy policy compliance through user perception: A case study of Chinese social media applications," *Data Sci. Inf.*, vol. 5, no. 1, pp. 1–17, Mar. 2025, doi: 10.1016/j.dsim.2025.05.003.

[21] G. Modalavalasa, "The Role of DevOps in Streamlining Software Delivery: Key Practices for Seamless CI/CD," *Int. J. Adv. Res. Sci. Commun. Technol.*, vol. 1, no. 12, pp. 258–267, Jan. 2021, doi: 10.48175/IJARSCT-8978C.

[22] A. Issaoui, J. Örtensjö, and M. S. Islam, "Exploring the General Data Protection Regulation (GDPR) compliance in cloud services: insights from Swedish public organizations on privacy compliance," *Futur. Bus. J.*, 2023, doi: 10.1186/s43093-023-00285-2.

[23] V. Prajapati, "Advances in Software Development Life Cycle Models : Trends and Innovations for Modern Applications," *J. Glob. Res. Electron. Commun.*, vol. 1, no. 4, pp. 1–6, 2025.

[24] O. Ajayi, "Data Privacy and Regulatory Compliance in the Usa: a Call for a Centralized Regulatory Framework," *Int. J. Sci. Res. Manag.*, vol. 12, no. 12, pp. 573–584, 2024, doi: 10.18535/ijsrm/v12i12.lla01.

[25] K. Olson and E. Keller, "Federating trust: Network orchestration for cross-boundary zero trust," in *Proceedings of the 2021 SIGCOMM 2021 Poster and Demo Sessions, Part of SIGCOMM 2021*, 2021. doi: 10.1145/3472716.3472865.

[26] R. Patel and P. B. Patel, "The Role of Simulation & Engineering Software in Optimizing Mechanical System Performance," *TIJER – Int. Res. J.*, vol. 11, no. 6, pp. 991–996, 2024.

[27] V. O. Nyangaresi, "Masked Symmetric Key Encrypted Verification Codes for Secure Authentication in Smart Grid Networks," in *Proceedings - 2022 IEEE 4th Global Power, Energy and Communication Conference, GPECOM 2022*, 2022. doi: 10.1109/GPECOM55404.2022.9815718.

[28] J. A. J. Alsayaydeh, Irianto, M. F. Ali, M. N. M. Al-Andoli, and S. G. Herawan, "Improving the Robustness of IoT-Powered Smart City Applications Through Service-Reliant Application Authentication Technique," *IEEE Access*, vol. 12, pp. 19405–19417, 2024, doi: 10.1109/ACCESS.2024.3361407.

[29] A. Kamruzzaman, S. Ismat, J. C. Brickley, A. Liu, and K. Thakur, "A Comprehensive Review of Endpoint Security: Threats and Defenses," in *2022 International Conference on Cyber Warfare and Security, ICCWS 2022 - Proceedings*, 2022. doi: 10.1109/ICCWS56285.2022.9998470.

[30] C. C. Cantarelli, B. Flybjerg, E. J. E. Molin, and B. van Wee, "Cost Overruns in Large-Scale Transport Infrastructure Projects," *Autom. Constr.*, 2018.

[31] B. S. Vidhyasagar, M. Arvindhan, A. Arulprakash, K. S. B. Bharathi, and S. Kalimuthu, "The Crucial Function that Clouds Access Security Brokers Play in Ensuring the Safety of Cloud Computing," in *2023 International Conference on Communication, Security and Artificial Intelligence, ICCSAI 2023*, 2023. doi: 10.1109/ICCSAI59793.2023.10420940.

[32] Y. Colomb, P. White, R. Islam, and A. Alsadoon, "Applying Zero Trust Architecture and Probability-Based Authentication to Preserve Security and Privacy of Data in the Cloud," in *Emerging Trends in Cybersecurity Applications*, 2022. doi: 10.1007/978-3-031-09640-2_7.

[33] V. Nagamalla, J. R. karkee, and R. K. Sanapala, "Integrating Predictive Big Data Analytics with Behavioral Machine Learning Models for Proactive Threat Intelligence in Industrial IoT Cybersecurity," *Int. J. Wirel. Ad Hoc Commun.*, 2023, doi: 10.54216/ijwac.070201.

[34] H. Kali and G. Modalavalasa, "Artificial Intelligence (AI)-Driven Business Intelligence for Enhancing Retail Performance with Customer Insights," *Asian J. Comput. Sci. Eng.*, vol. 9, no. 4, pp. 1–9, 2024.

[35] M. L. Gambo and A. Almulhem, "Zero Trust Architecture: A Systematic Literature Review," 2025.

[36] J.-H. Park, S.-C. Park, and H.-Y. Youm, "A Proposal for a Zero-Trust-Based Multi-Level Security Model and Its Security Controls," *Appl. Sci.*, vol. 15, no. 2, p. 785, Jan. 2025, doi: 10.3390/app15020785.

[37] P. Dhiman *et al.*, "A Review and Comparative Analysis of Relevant Approaches of Zero Trust Network Model," *Sensors*. 2024. doi: 10.3390/s24041328.

[38] B. D. Lund, T. Lee, Z. Wang, and T. Wang, "Zero Trust Cybersecurity : Procedures and Considerations in Context," pp. 1–14, 2024.

[39] W. Yeoh, M. Liu, M. Shore, and F. Jiang, "Zero trust cybersecurity: Critical success factors and A maturity assessment framework," *Comput. Secur.*, vol. 133, p. 103412, Oct. 2023, doi: 10.1016/j.cose.2023.103412.

[40] D. Patel, "Zero Trust and DevSecOps in Cloud-Native Environments with Security Frameworks and Best Practices," *Int. J. Adv. Res. Sci. Commun. Technol.*, vol. 3, no. 3, 2023.