

A Minimal Overlay-Based Framework for Transitioning Legacy Infrastructure to Zero Trust

Wenjia Wang¹, Seyed Masoud Sadjadi², Naphtali Rishen³, Arpan Mahara⁴

Knight Foundation School of Computing and Information Sciences

Florida International University

Miami, USA

Email: wwang048@fiu.edu¹, sadjadi@cs.fiu.edu², rishen@cs.fiu.edu³, amaha038@fiu.edu⁴

Abstract— Traditional perimeter-based security models struggle to secure legacy systems against evolving threats posed by remote work, IoT adoption, and cloud migration. Yet most Zero Trust (ZT) roadmaps demand disruptive refactoring that many organizations cannot afford. We present a lightweight, identity-centric transition model that overlays, rather than replaces, existing networks, relying on just three open-source components: Identity & Access Management, Public Key Infrastructure, and Continuous Diagnostics & Mitigation. A three-node Azure prototype using StrongSwan mutual TLS tunnels demonstrates that the full control-plane bundle idles at approximately 8% CPU and 320 MB RAM, each endpoint agent under 1% CPU and 30 MB RAM, and encrypted throughput remains within 2.5% of underlay performance while certificate revocation propagates in 8 min 14 s. These results show that meaningful ZT protections can be deployed immediately—no new hardware, rewiring, or licensing—offering a practical path to deeper ZT maturity.

Index Terms— Cybersecurity, Zero Trust, Transition, Overlay Network, Policy Enforcement Point, Access Control

I. INTRODUCTION

The traditional perimeter-based security model is no longer sufficient. Today's enterprise environments are shaped by cloud migration, remote work, Bring-Your-Own-Device (BYOD) policies, and a surge in unmanaged IoT endpoints, all of which blur network boundaries and expose legacy systems to credential theft and lateral movement [1]. In response to these evolving threats, regulators worldwide, including the U.S. Office of Management and Budget, now mandate a Zero Trust Architecture (ZTA) posture for government agencies by 2027 [2] [3].

Zero Trust (ZT) rethinks security from the ground up. It assumes breaches are inevitable and enforces authentication, authorization, and continuous monitoring for every access request, regardless of network location. However, ZT adoption faces serious obstacles in legacy environments.

1) *Problem:* ZT adoption in legacy environments is hindered by three key challenges: First, frameworks often lack actionable implementation guidance [4]; second, static infrastructures resist dynamic policy enforcement [5]; and third, full system replacement is often infeasible due to cost and complexity [6].

To address this, we propose a minimal, identity-centric overlay framework that integrates with existing networks. It provides practical tools and steps to realize ZT principles with low disruption.

Our work is guided by two questions: What is the smallest viable component set for ZT? How can these components deliver immediate gains in brownfield deployments?

2) *Contribution:* This research offers the following key contributions:

- **Framework:** A threat-model-driven ZT transition model tailored for legacy systems.
- **Prototype:** A working Azure-based implementation using IAM, PKI, and CDM.
- **Evaluation:** Empirical results on performance, revocation latency, and deployment overhead.

II. RELATED WORK

Kindervag et al. introduced ZT emphasizing continuous verification beyond traditional perimeters [7]. NIST's SP 800-207 formalized key ZT principles such as separation of control and data planes [8], while CISA's ZT Maturity Model [3] and DoD's ZT Overlays [9] guide large-scale deployments. However, these frameworks typically demand substantial resources, challenging small and medium enterprises (SMEs). Our approach simplifies deployment, concentrating specifically on Identity & Access Management (IAM), Public Key Infrastructure (PKI), and Continuous Diagnostics & Mitigation (CDM).

Existing migration strategies include perimeterless platforms like BeyondCorp [10] and Cloudflare One [11], which use global proxies and centralized policy enforcement. Though effective, these models entail extensive infrastructure and vendor lock-in. Open-source alternatives such as OpenZiti [12] and ZeroTier [13] utilize encrypted identity-based mesh overlays but lack detailed empirical validations on operational metrics. Similarly, academic frameworks [14] [15] propose phased migrations without complete prototypes or extensive validations. Our reproducible prototype fills these gaps by providing empirical performance data.

Identity-centric ZT solutions include SPIFFE's standardized short-lived X.509 certificates [16] and Netflix's BLESS

SSH-based ephemeral identities [17], both relying heavily on external enforcement. In contrast, our solution supports practical identity provisioning and continuous verification specifically tailored to minimally modified legacy systems.

Evaluation methods such as STRIDE [18] and OWASP’s Threat Modeling Cheat Sheet [19] offer structured threat analysis. Empirical frameworks by Basta et al. [20] and Capili [21] provide methodologies for evaluating micro-segmentation effectiveness and resilience in IoT scenarios. Motivated by these works, we present measurable revocation latency and empirical validation through a reproducible three-VM prototype.

III. THREAT MODEL AND ASSUMPTIONS

Our analysis aligns with CISA’s ZT Maturity Model [3] and DoD’s ZT Overlays [9], cross-checked against threat models from Google BeyondCorp [10] and OpenZiti [12].

1) Assets:

- Control-plane services: IAM/PKI controller VMs, policy database, certificate-revocation endpoints.
- Data-plane endpoints: user devices, legacy servers, resource nodes that expose file shares.
- Secrets: X.509 private keys, short-lived tokens, StrongSwan Pre-Shared Keys (PSKs).
- Telemetry: posture reports and CDM logs retained for 30 days.

2) Adversary Model:

- External Advanced Persistent Threat (APT) with full network visibility, able to replay or inject packets.
- Malicious insider with valid corporate credentials and physical access to one managed device.
- Compromised endpoint (malware or remote-code execution) trying to pivot laterally.

We assume adversaries have ample time and moderate funding but no direct access to the Azure hypervisor.

3) Assumptions:

- Azure IaaS provides honest-but-curious availability; root-of-trust hardware is outside scope.
- Clocks are synchronized within ± 30 seconds via Network Time Protocol (NTP) for accurate certificate validation.
- TLS 1.3 libraries are patched at the OS level.
- Users enroll devices through an attacker-resistant out-of-band channel.

4) *Attack Surface and Threat Enumeration:* Table I summarizes the key attack surfaces mapped to CISA ZT maturity goals, and Table II outlines the threat mappings aligned with OWASP’s threat-modeling categories [19].

5) Security Objectives:

- 1) **Authenticated entry:** Only nodes presenting a valid, non-revoked certificate issued by our CA may establish a tunnel.

TABLE I
ATTACK SURFACE ENUMERATION

Flow / Interface	Principal Threats	Notes
Identity Enrollment API	Spoofing, Tampering	Bootstrap credential reuse, CSR manipulation
mTLS Handshake	Downgrade, replay	Forced downgrade to a weaker cipher and exposure to a stale-token window
Overlay Data Channel	Traffic capture, injection	Side-door into file transfer
Certificate Status	Info disclosure, DoS	Blocking revocation checks stalls auth
Management CLI / REST	Elevation, repudiation	Admin token theft, log wiping

TABLE II
STRIDE THREAT MAPPING AND MITIGATIONS

Asset	S	T	R	I	D	E	Mitigations
Control-plane API	✓						mTLS + RBAC; audit logs
Endpoint cert store		✓	✓	✓			Encrypted disk; short-lived certs
Tunnel handshake	✓	✓		✓	✓		TLS 1.3, AEAD, anti-replay
CDM telemetry			✓			✓	Hash chaining; rate-limit

- 2) **Rapid containment:** A compromised identity can be revoked and fully blocked across the overlay within 10 minutes.
- 3) **Least privilege:** Policy Enforcement Points ensure an endpoint reaches only the specific resource nodes authorized by IAM.
- 4) **No silent lateral movement:** CDM alerts on attempts by endpoints to access unauthorized peers, even via established tunnels.

IV. PROPOSED TRANSITION MODEL

This section delivers the paper’s core contribution: a reproducible pathway from perimeter-based security to identity-driven ZTA. Our model operationalizes three widely endorsed principles:

- 1) **Identity Verification:** Every user, device, and service must present verifiable, continuously validated credentials, embodying “never trust, always verify”.
- 2) **Policy Enforcement:** Fine-grained, least-privilege access is applied at every request, ensuring entities receive only what they need—no more, no less.
- 3) **Continuous Monitoring:** Real-time telemetry enables rapid detection of threats and automatic response to deviations, fulfilling “assume breach”.

A. Fundamental Technologies

Our transition model relies on five lightweight building blocks that bring ZT principles to legacy networks without new hardware or proprietary stacks:

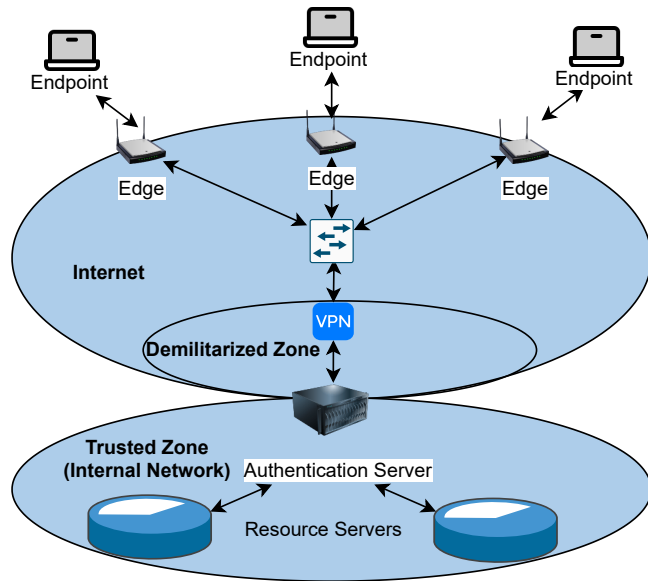


Fig. 1. Traditional Network Security Architecture

- 1) **Overlay Network:** A cryptographically protected mesh overlay decouples security from the underlay, so legacy hosts join without re-addressing.
- 2) **Identity:** Every user, device, and service presents a unique, verifiable identity, eliminating implicit trust in network location [22].
- 3) **Public Key Infrastructure (PKI):** A trusted CA issues short-lived X.509 certificates that bind keys to identities and support rapid revocation [23], which is foundational for “never trust, always verify.”
- 4) **Mutual TLS (mTLS):** Client and server mutually authenticate with their certificates before any data flows, ensuring confidentiality, integrity, and proof of identity on every connection [24].
- 5) **Identity and Access Management (IAM):** IAM maps certified identities to least-privilege policies and enforces them at each request, providing continuous authorization [25].

B. Design

Our design replaces predefined “trusted” segments with a unified, node-centric overlay, deliberately limited to IAM, PKI, and CDM to provide a minimal viable ZT solution for legacy servers.

1) *Abstraction Through Nodes:* Traditional networks (Fig. 1) rely on trust-based segmentation, complicating security management. Our model (Fig. 2) replaces implicit trust with identity-verified nodes in a unified overlay, separating responsibilities into control and data planes as per NIST guidelines [8].

2) *Control Plane:* To ensure dynamic and responsive security management, our model designates specific nodes as “control nodes” hosting critical security components:

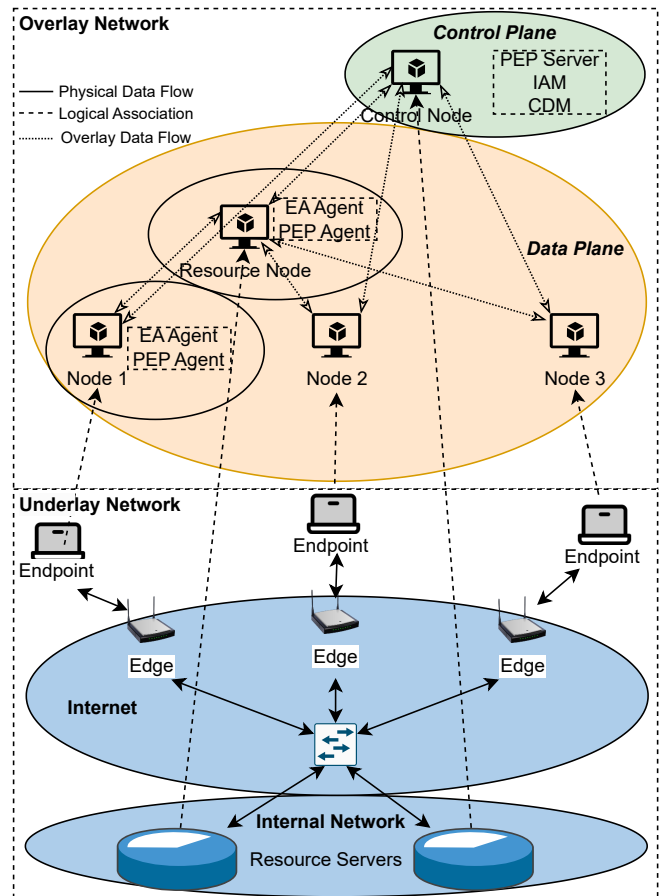


Fig. 2. Proposed Zero Trust Transition Model

- Policy Enforcement Point (PEP) Server: Manages and enforces granular access policies.
- IAM: Performs identity verification and authorization decisions.
- CDM: Continuously evaluates the security posture of all nodes, alerting IAM and PEP Server to security anomalies.

Components can be co-located for small deployments or distributed across nodes for scalability and reliability.

3) *Data Plane:* All other devices in the overlay belong to the data plane. Each node is equipped with two specifically designed agents:

- Endpoint Access (EA) Agent: Manages initial enrollment into the overlay network, obtaining its identity.
- PEP Agent: Enforces real-time policy decisions at the endpoint level, acting immediately on instructions from the control plane.

All Data Plane nodes, including Resource and Endpoint Node 1–3 in Fig. 2, are logically equal.

4) *Topology:* Crucially, endpoints do not need tunnels to each other; they communicate directly with resource nodes and the control plane. Resource nodes, as specialized endpoints, handle inbound secure connections from the endpoints

needing access. The control plane also establishes or tears down tunnels as policies change or device posture updates occur.

C. Interaction Flow and Alignment with CISA ZT Maturity Model

Fig. 3 captures the end-to-end flow. An endpoint first enrolls with IAM to obtain a short-lived X.509 identity and then establishes an mTLS tunnel validated by PKI and enforced by the PEP Server. All application traffic traverses this overlay, while posture telemetry from PEP agents streams to CDM. IAM consumes these signals and issues policy updates that the PEP Server applies instantly, ensuring continuous authentication, authorization, and risk-adaptive access—without touching the underlay. Table III shows our minimal stack mapped to CISA’s ZT Maturity Model pillars.

TABLE III
PROTOTYPE COVERAGE VS. CISA ZTMM PILLARS

Pillar	Covered Component(s)	Maturity
Identity	PKI, IAM	Advanced
Device	CDM health checks	Intermediate
Network / Env.	StrongSwan overlay, PEP	Advanced
Application	—	Not targeted
Data	—	Not targeted
Visibility	CDM logs	Initial
Automation	Bash scripts, GitHub CI	Initial
Governance	HR role feed → IAM	Intermediate

V. IMPLEMENTATION

We validated our ZT transition framework using a minimal, representative setup comprising three Azure virtual machines (VMs). Each node has a distinct role:

- Control Node: Manages IAM and PKI for identity, certificate, and policy handling.
- Resource Node: Provides resources, enforces access policies.
- Endpoint Node: Represents user devices requesting resources.

All nodes were deployed in one Azure subnet to simulate an organization’s internal network transitioning to ZT. Experimental steps are detailed below:

- 1) **Infrastructure Setup:** Deployed three Ubuntu 22.04 VMs in a dedicated Azure resource group and subnet.
- 2) **IAM & PKI Setup:**
 - Developed a Flask-based IAM integrated with PKI for credential authentication, ephemeral tokens, and X.509 certificate issuance.
 - PKI assigns unique identities via certificates.
- 3) **Node Identity Enrollment:**
 - a) Defined username-password pairs within IAM.
 - b) Nodes authenticate and request ephemeral tokens using credentials.
 - c) Nodes generate key pairs locally, send Certificate Signing Requests (CSRs) with tokens.

- d) IAM validates requests and issues signed X.509 certificates.

- 4) **Role-Based Certificate Management:** IAM periodically polls the user directory and triggers revocation workflows upon detecting terminated roles.

- 5) **Overlay Network Setup:**

- a) Implemented overlay network using StrongSwan for simplicity, modularity, and legacy integration.
- b) StrongSwan establishes authenticated, secure tunnels via X.509 certificates:
 - Control–Endpoint Tunnel: policy and certificate management.
 - Resource–Control Tunnel: policy and certificate management.
 - Resource–Endpoint Tunnel: secure resource access.

- 6) **Policy Enforcement Setup:** We implemented a simplified PEP agent and CDM simulation directly on the Resource Node rather than centrally on the Control Node to prioritize clarity and simplicity in our proof-of-concept:

- **PEP Policies:**

- a) Deny all communication without authenticated tunnels.
- b) Dynamically permit communication upon tunnel establishment.

- **CDM Functionality:**

- Monitors tunnel status.
- Signals PEP to enforce policies dynamically based on tunnel activity.

- 7) **Endpoint Node Setup:** EA and PEP functionalities implemented via two shell scripts:

- `connect.sh`: Verifies policy compliance as a PEP, then initiates an mTLS tunnel as an EA.
- `disconnect.sh`: Terminates the tunnel and re-applies restrictive policies, fulfilling the PEP role.

VI. EVALUATION

To evaluate functional correctness of our model, we implemented a simple file-sharing application to conduct tests.

A. Experimental Setup

We deployed our evaluation environment on three Azure VMs. Environment details are in Table IV. Clocks synchronized within ± 30 s via NTP satisfy assumptions in Section IV-C.

B. Functional Correctness

We repeated the file-transfer test in two scenarios:

- 1) **Overlay Tunnel Active:** We activated the Endpoint Node’s overlay tunnel via `connect.sh`. A subsequent attempt to transfer a file succeeded, which demonstrated effective policy enforcement.

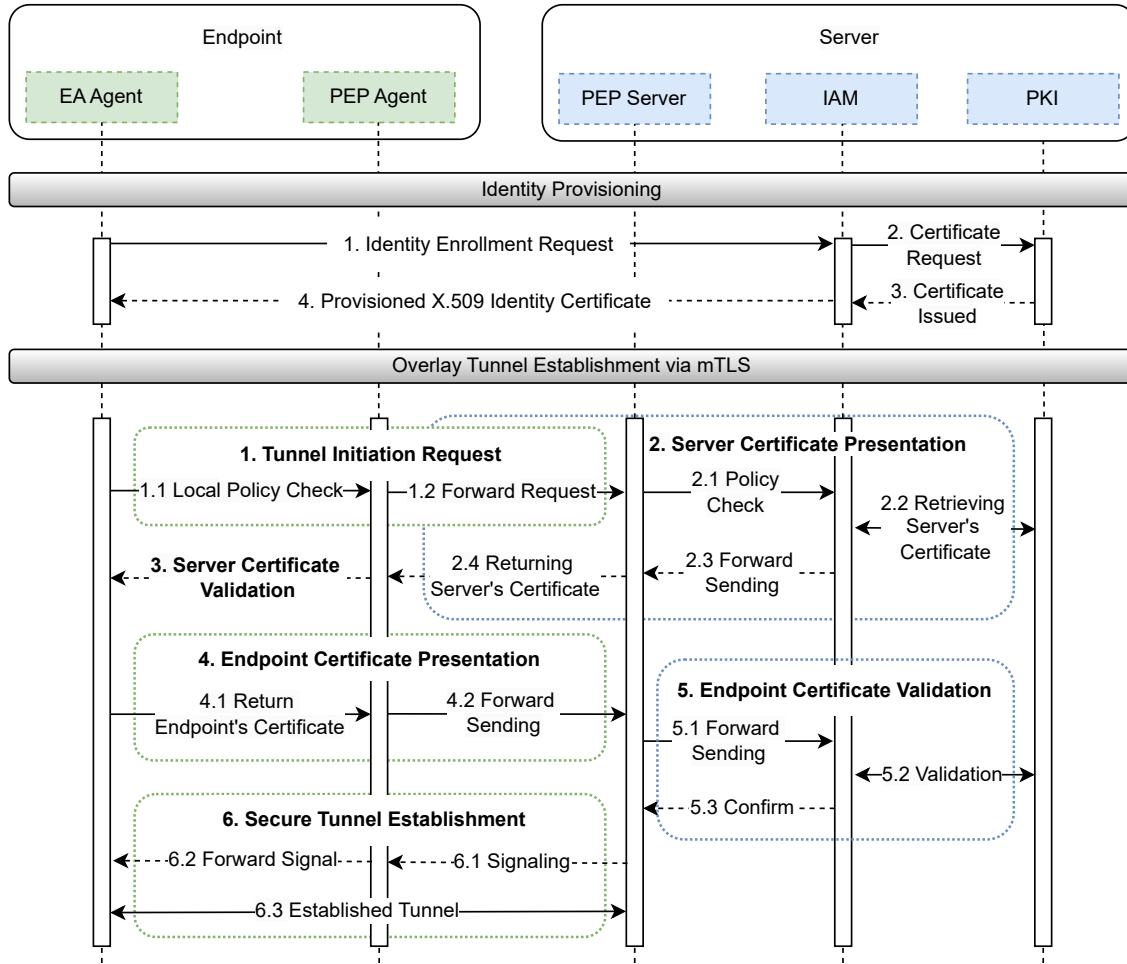


Fig. 3. Identity Provisioning & Overlay Network Tunnel Establishment

 TABLE IV
 EXPERIMENTAL CONFIGURATION (AZURE VMs)

Item	Value
Cloud Platform	Microsoft Azure
Number of VMs	3 (Control, Resource, Endpoint)
VM Size	Standard B2s (2 vCPUs, 4 GiB RAM each)
OS / Kernel	Ubuntu Server 22.04 LTS / Kernel 5.15
iperf3 Version	3.13
Overlay Tunnel	StrongSwan with TLS 1.3 certificates
Testing Scenario	5 × 10 s TCP, default window, single stream
Time Synchronization	NTP, ±30 s accuracy

- 2) Overlay Tunnel Inactive: After disabling the Endpoint's overlay tunnel, file transfer failed, confirming policy enforcement prevents unauthorized connections.

These scenarios validate Objectives 1 and 3, confirming that resource access requires explicit authentication.

C. Performance Benchmarks

Overlay performance overhead was measured by five 10-second TCP iperf3 tests between Endpoint and Resource

nodes via mTLS tunnels. Table V summarizes the measured throughput and retransmissions.

The overlay achieved an average throughput of 92.9 Mbps, only 2.5% lower than the direct underlay (95.3 Mbps). No retransmissions occurred, confirming minimal overhead.

 TABLE V
 OVERLAY TUNNEL PERFORMANCE: FIVE CONSECUTIVE RUNS

Run	1	2	3	4	5	Mean
Throughput (Mbps)	92.8	93.2	92.5	92.9	93.1	92.9
Retransmits	0	0	0	0	0	0

D. Adversarial Tests

We further validated overlay robustness against common adversarial actions:

- **Replay Attack Test** (Objective 1): Captured IKE_SA_INIT handshake packets replayed from spoofed IP addresses were correctly rejected ("invalid IKE_SA"), confirming active replay protection.
- **Certificate Revocation Test** (Objective 2): We revoked an active endpoint certificate via OCSP during an ongo-

ing session. The overlay tunnel terminated successfully within 8 min 14 s, well under the target 10-minute threshold. This rapid revocation ensures compromised identities are quickly contained, aligning with CISA's "Continuous Verification" guideline.

- **Unauthorized Lateral Movement Test (Objective 4):** To test Objective 4, we restricted the Endpoint to Resource Nodes via IAM policy. An unauthorized HTTP request to the Control Node was denied and logged; real-time CDM alerting remains future work.

E. Operational Overhead

Our model uses lightweight agents and a single co-located control-plane, minimizing resource demands. IAM, PKI, CDM, and PEP server share one small VM (2 vCPUs, 4 GB RAM), consuming 8% CPU and 320 MB RAM at idle. Each enrolled endpoint runs a combined PEP/EA agent consuming less than 1% CPU and approximately 30 MB RAM in steady state. Deployment requires no new hardware, rewiring, or recompilation; endpoints join via an approximately 20-second bootstrap script. As all components are open-source, our approach introduces no licensing or subscription costs.

All scripts used for implementation and tests are available in our GitHub repository [26].

VII. CONCLUSION AND FUTURE WORK

We introduced and validated a minimal ZT transition framework deployable on legacy infrastructure with minimal operational overhead. By focusing on IAM, PKI, and CDM, the design reduces complexity while meeting CISA's Intermediate maturity for Identity and Network pillars. Experiments confirmed identity-based authentication, rapid certificate revocation, and least-privilege enforcement at 92.9 Mbps throughput with only 2.5% overhead on standard virtual hardware. The open-source implementation and automation scripts offer a practical starting point for securing legacy environments.

Future work includes standardized benchmarks for latency and overhead, acceleration of certificate revocation, and leveraging AI-based CDM analytics for adaptive policy enforcement.

ACKNOWLEDGMENT

This material is based in part upon work supported by the National Science Foundation under Grant No. MRI20 CNS-2018611.

REFERENCES

- [1] W. S. Admass, Y. Y. Munaye, and A. A. Diro, "Cyber security: State of the art, challenges and future directions," *Cyber Security and Applications*, vol. 2, p. 100031, 2024.
- [2] Office of Management and Budget, "Moving the U.S. Government Toward Zero Trust Cybersecurity Principles," Memorandum M-22-09, Jan. 26, 2022. [Online]. Available: <https://www.whitehouse.gov/wp-content/uploads/2022/01/M-22-09.pdf> [Accessed Jun. 10, 2025].
- [3] U.S. Cybersecurity and Infrastructure Security Agency (CISA), "Zero Trust Maturity Model," Version 2.0, 2023.
- [4] N. F. Syed, S. W. Shah, A. Shaghaghi, A. Anwar, Z. Baig and R. Doss, "Zero Trust Architecture (ZTA): A Comprehensive Survey," in *IEEE Access*, vol. 10, pp. 57143-57179, 2022.
- [5] S. Teerakanok, T. Uehara, and A. Inomata, "Migrating to zero trust architecture: Reviews and challenges," *Security and Communication Networks*, vol. 2021, no. 1, p. 9947347, 2021.
- [6] A. Miller, "Zero Trust Architecture: Implementation and Challenges," AgileBlue, May 1, 2024. [Online]. Available: <https://agileblue.com/zero-trust-architecture-implementation-and-challenges/>. [Accessed: Sep. 7, 2024].
- [7] J. Kindervag, "Build Security Into Your Network's DNA: The Zero Trust Model," Forrester Research, Apr. 2010.
- [8] NIST, "Zero Trust Architecture," NIST Special Publication 800-207, Aug. 2020. [Online]. Available: <https://doi.org/10.6028/NIST.SP.800-207> [Accessed: May 10, 2025].
- [9] R. Freter, "Department of Defense Zero Trust Reference Architecture, Version 2.0," Dept. of Defense CIO, Jul. 2022. [Online]. Available: <https://dodcio.defense.gov/Portals/0/Documents/Library/ZeroTrustOverlays.pdf> [Accessed: May 1, 2025].
- [10] R. Ward and B. Beyer, "BeyondCorp: A new approach to enterprise security," *USENIX; login.*, vol. 39, no. 6, pp. 6-11, Dec. 2014.
- [11] Cloudflare, "Cloudflare One: More than just a VPN replacement," Cloudflare Blog, Oct. 2020. [Online]. Available: <https://blog.cloudflare.com/cloudflare-one/> [Accessed: May 2, 2025].
- [12] OpenZiti, "OpenZiti Documentation," NetFoundry, 2025. [Online]. Available: <https://openziti.io/docs/learn/introduction/> [Accessed: May 4, 2025].
- [13] ZeroTier Inc., "ZeroTier Networking," 2025. [Online]. Available: <https://www.zerotier.com> [Accessed: June 1, 2025].
- [14] P. Phiayura and S. Teerakanok, "A comprehensive framework for migrating to zero trust architecture," *IEEE Access*, vol. 11, pp. 19487-19511, 2023.
- [15] S. Xiao, Y. Ye, N. Kanwal, T. Neue, and B. Lee, "SoK: Context and Risk Aware Access Control for Zero Trust Systems," *Security and Communication Networks*, vol. 2022, Art. ID 7026779, 2022.
- [16] CNCF, "SPIFFE: Secure Production Identity Framework For Everyone," Cloud Native Computing Foundation. [Online]. Available: <https://spiffe.io/>. [Accessed: Jun. 1, 2025].
- [17] Netflix, "Netflix BLESS: SSH Certificate Authority," GitHub Repository. [Online]. Available: <https://github.com/Netflix/bless>. [Accessed: Jun. 2, 2025].
- [18] Microsoft, "STRIDE Threat Modeling," Microsoft Security Development Lifecycle, 2023. [Online]. Available: <https://learn.microsoft.com/security/engineering/stride-threat-modeling> [Accessed: May 6, 2025].
- [19] OWASP Foundation, "Threat Modeling Cheat Sheet," 2023. [Online]. Available: https://cheatsheetsseries.owasp.org/cheatsheets/Threat_Modeling_Cheat_Sheet.html [Accessed: May 7, 2025].
- [20] N. Basta, M. Ikram, M. A. Kaafar, and A. Walker, "Towards a Zero-Trust Micro-segmentation Network Security Strategy: An Evaluation Framework," in *Proc. IEEE/IFIP NOMS*, Budapest, Hungary, 2022, pp. 1-7.
- [21] M. Capili, "Simulation-Based Evaluation of Perimeter-Based and Zero Trust Security Implementation on Internet of Things," Ph.D. dissertation, The George Washington Univ., Washington, DC, 2024.
- [22] National Institute of Standards and Technology, "Digital Identity Guidelines," NIST Special Publication 800-63-3, Gaithersburg, MD, USA, Jun. 2017.
- [23] D. Cooper *et al.*, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile," Internet Engineering Task Force, RFC 5280, May 2008.
- [24] ISO, "IEC 29115: 2013—Information Technology—Security Techniques—Entity Authentication Assurance Framework," 2013.
- [25] E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.3," Internet Engineering Task Force, RFC 8446, Aug. 2018.
- [26] W. Wang, "ZeroTrust_Transition_Codebase," GitHub. [Online]. Available: https://github.com/Wenjia1215/ZeroTrust_Transition_Codebase. [Accessed: Jul. 23, 2025].