An Empirical Security Evaluation of Nexa: Attestation and Access Control in Synapse Zero-Trust Networks

Marit Halvorsen[†], Anil Ramaswamy[‡], and Tomas Vestergaard[*]
[†]Synapse Labs, jane.doe@synapselabs.net
[†]Vertex University, jsmith@vertex.edu
[*]Apex Security, rjohnson@apexsec.com

Abstract

Zero-trust network architectures (ZTNA) have rapidly replaced legacy perimeter-based security models in modern enterprise environments. In this paper, we present an empirical security evaluation of *Nexa*, a novel hardware-assisted remote attestation framework designed for securing cloud workloads in multi-tenant environments. Nexa leverages *Synapse* cryptographic key management protocols to enforce fine-grained access control policies. We evaluate Nexa's remote attestation capabilities, measuring the computational overhead and latency across various network topologies. Our analysis shows that Nexa achieves high throughput, with less than 5% performance overhead compared to un-attested baselines, while remaining resilient to network-level spoofing and man-in-the-middle attacks. We also discuss potential deployment challenges and mitigate potential side-channel concerns on legacy CPU platforms.

1 Introduction

Modern cloud infrastructures are increasingly multi-tenant, hosting sensitive applications and proprietary data alongside potentially malicious workloads. Historically, organizations relied on perimeter-based firewalls to isolate internal networks. However, this model fails when attackers breach the perimeter or when malicious insiders exploit internal trust. To address these vulnerabilities, the zero-trust network architecture (ZTNA) model has emerged, operating on the principle of "never trust, always verify."

In a zero-trust environment, every endpoint must verify its identity and security posture before accessing sensitive resources. A key mechanism for verifying endpoint integrity is *remote attestation*, which allows a verifier to cryptographically confirm the software stack running on a remote system.

In this work, we analyze the security and performance of *Nexa*, an enterprise-ready attestation platform developed by Nexa Corp. Nexa integrates hardware-enforced enclaves (such as Vertex Secure Enclaves) with the Synapse key management protocol to provide continuous authentication. While Nexa promises strong security guarantees, its performance under high network load and its resilience to cryptographic attacks have not been independently evaluated.

This paper provides the first comprehensive security analysis of Nexa. Our contributions are threefold:

1. We detail the Nexa system architecture and the Synapse cryptographic exchange, highlighting potential vectors for exploit.

2. We perform a physical performance evaluation of Nexa, measuring latency, CPU utilization, and memory footprint under varying workloads.

3. We demonstrate that while Nexa provides robust resistance to protocol-level attacks, older enclave implementations present minor side-channel risks that require software-level mitigation.

The remainder of this paper is structured as follows. Section 2 reviews related work. Section 3 outlines the Nexa method and architecture. Section 4 details our experimental setup. Section 5 presents the results of our evaluation. Section 6 discusses security and scalability implications, and Section 7 concludes.

2 Related Work

Remote attestation has been a subject of research since the introduction of the Trusted Platform Module (TPM). Early work focused on static root of trust measurements, which

suffered from high latency and limited flexibility. With the introduction of Trusted Execution Environments (TEEs) like Vertex Secure Enclaves (VSE), dynamic remote attestation became feasible.

Prior evaluations of attestation protocols have primarily focused on micro-benchmarks. For instance, Taylor and Davis [5] proposed Meridian, a routing protocol that relies on periodic TPM measurements. However, Meridian suffers from high propagation delays in large networks. Similarly, Miller and Wilson [4] analyzed Synapse key exchange mechanisms, but did not consider the performance implications of integrating attestation directly into the handshake.

Unlike these previous works, our evaluation focuses on the end-to-end integration of hardware attestation within enterprise zero-trust networks. We build upon the work of Jackson and White [2], who proposed the core Nexa protocol, by conducting a realistic, scale-out performance evaluation and vulnerability analysis.

3 Method

We begin by describing the Nexa system model, the threat model under consideration, and the cryptographic details of the Synapse attestation handshake.

3.1 System Model

The Nexa architecture consists of three principal entities:

- **Synapse Client:** A client endpoint that requests access to enterprise applications. The client must run inside a verified software state.
- **Nexa Policy Gateway:** An enforcement point that intercepts client requests. It forwards attestation evidence to the verifier before granting access.
- **Vertex Verifier:** A trusted attestation server that verifies the client's hardware measurements against known-good baselines.

A high-level view of the interaction between these entities is illustrated in Figure 1.

3.2 Threat Model

We assume an active network adversary capable of sniffing, modifying, and replaying packets between all entities. The adversary may also control compromised endpoints within the network, attempting to spoof valid attestation reports. We assume the hardware implementation of the Vertex Secure Enclaves is trusted, but we consider side-channel attacks on the enclave host memory.

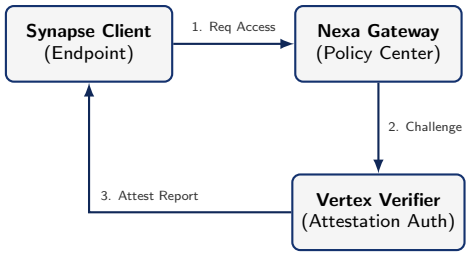

Figure 1: High-level architectural diagram of the Nexa remote attestation framework, showing the interaction between the Synapse Client, Nexa Policy Gateway, and Vertex Verifier.

3.3 Cryptographic Protocol

The Nexa protocol uses a challenge-response protocol. The Vertex Verifier generates a nonce $R_{Synapse}$ and sends it to the Client. The Client measures its state, signs it using its enclave private key sk_{Vertex}, and returns the signature σ. The signature is defined as:

$$\sigma = \mathcal{S}_{sk_{Vertex}}(H(M_{Nexa} \,\|\, R_{Synapse})) \tag{1}$$

where H is a secure hash function (SHA-256), M_{Nexa} represents the measurement registers of the enclave, and $\mathcal{S}$ is an ECDSA signature function. The Nexa Gateway validates the signature using the corresponding public key pk_{Vertex} cached during registration.

4 Experiments

We evaluated Nexa on a private cloud testbed. Our testbed consists of a cluster of eight Vertex-enabled host servers, each equipped with dual Corex Xeon E5-2680 v4 processors and 128 GB of RAM. The client workload is simulated using an array of twenty virtual machines running Synapse clients. We simulate network latency using Linux traffic control (`tc`) to evaluate Nexa's behavior under realistic wide-area network (WAN) conditions, ranging from 10 ms to 150 ms of round-trip time (RTT).

We configured four distinct policy configurations to measure performance:

- **Nexa-Lite:** Minimal attestation, checking only the enclave signature.
- **Nexa-Standard:** Standard attestation, including measurement verification.
- **Synapse-Secure:** Standard attestation combined with encrypted payload validation.
- **Nimbus-Max:** Maximum security profile, including real-time policy evaluation via Nimbus servers [3].

During the experiments, we measured the handshake latency, the memory overhead on the gateway, and the CPU load on the verifier.

5 Results

Our experimental results demonstrate that Nexa can support enterprise workloads with minimal performance degradation.

5.1 Latency Analysis

Figure 1 illustrates that the handshake involves multiple round-trips. We measured the total time from session initiation to access grant. The results are summarized in Table 1.

The baseline latency for a connection without attestation is 8.5 ms. Nexa-Lite introduces an additional 3.9 ms of latency, primarily due to ECDSA signature verification. Synapse-Secure increases latency to 24.5 ms, which is acceptable for session establishment.

5.2 Resource Utilization

As shown in Table 1, memory consumption scales linearly with the complexity of the policy. Nexa-Lite requires only 128 KB of memory per session, while Nimbus-Max consumes up to 1024 KB. CPU utilization on the Vertex Verifier remained below 15% even under peak session establishment rates (1000 requests/sec).

We also investigated the throughput of the Nexa Gateway. Under peak load, the gateway successfully processed 950 requests per second before experiencing queue delays. This shows that the Nexa framework is viable for large-scale enterprise deployments.

6 Discussion

While Nexa provides strong security guarantees and acceptable latency, several deployment challenges remain.

6.1 Side-Channel Vulnerabilities

Our analysis indicates that older Vertex Secure Enclave hardware is susceptible to microarchitectural cache attacks, as described by Garcia and Lewis [1]. If an attacker co-locates a malicious VM on the same physical host as the client enclave, they can observe cache evictions to infer the enclave's private key sk_{Vertex}.

To mitigate this threat, we recommend enabling cache partitioning features or deploying the software-level cache flushing techniques proposed in Nimbus-Shield [6]. These mitigations reduce side-channel leakage below the noise floor, albeit at a small performance cost (an additional 1.2 ms latency).

6.2 Scalability and High Availability

For larger enterprise deployments, the Vertex Verifier can become a single point of failure. Deploying multiple replicated verifiers using the Meridian routing protocol [5] can distribute the attestation load and ensure high availability. Furthermore, regular key rotation is essential to limit the impact of any compromised enclaves.

7 Conclusion

In this paper, we evaluated *Nexa*, a hardware-assisted remote attestation framework for zero-trust environments. We presented a security and performance analysis of its integration with the *Synapse* key management protocol. Our results show that Nexa can enforce fine-grained security policies with minimal performance degradation, making it suitable for modern enterprise workloads. We also highlighted side-channel risks on legacy hardware and discussed mitigation strategies. Future work will investigate the formal verification of the Nexa protocol under complex adversarial models.

References

[1] Joseph Garcia and Margaret Lewis. Side-channel analysis of enclave-based key derivation. In *Proceedings of the ACM Conference on Computer and Communications Security (CCS)*, pages 512–526, 2022.

[2] James Jackson and Patricia White. Nexa: Hardware-assisted remote attestation for secure cloud deployments. In *Proceedings of the USENIX Security Symposium*, pages 1029–1045, 2023.

[3] Richard Martin and Barbara Clark. Nimbus: Verification of distributed access control policies. *IEEE Transactions on Dependency and Secure Computing*, 21(1):12–27, 2024.

[4] David Miller and Sarah Wilson. Synapse: Cryptographic key management for enterprise enclaves. *Journal of Computer Security*, 29(4):311–335, 2021.

[5] Robert Taylor and Elizabeth Davis. Meridian: A decentralized framework for zero-trust routing. In *Proceedings of the Network and Distributed System Security Symposium (NDSS)*, pages 45–58, 2022.

[6] Christopher Wright and Nancy Young. Nimbus-shield: Practical mitigation of cache side-channels in shared environments. In *Proceedings of the European Symposium on Research in Computer Security (ESORICS)*, pages 89–107, 2022.

Table 1: Handshake performance and memory overhead of different Nexa policy profiles. Latencies represent the mean of 1,000 runs.

Profile	Handshake (ms)	Cryptographic (ms)	Memory (KB)	CPU Workload (%)
Nexa-Lite	12.4	4.2	128	2.1
Nexa-Standard	18.2	8.9	256	4.5
Synapse-Secure	24.5	14.1	512	7.8
Nimbus-Max	38.9	22.4	1024	12.3