Journal of ICT Aplication and System (JICTAS)
Volume 5 Number 1, Juni 2026, pp. 1 ~10
ISSN: 2830-098X, DOI: 10.56313/jictas.v5i1.514

SYNTHTM: Continuous, AI-Driven Threat Modeling for Software Supply Chain Risk Propagation

Hilmand Khan[1], Saim Sujjad[2], Adan Raza Masoom[3], Riyan Rehman[4]
[1,2]Computer Science, Air University Islamabad, Pakistan

Article Info

Article history:

Received 04 25, 2026
Revised 05 08, 2026
Accepted 06 06, 2026

Keywords :

Software Supply Chain Security, Threat Modeling Graph Neural Networks, Large Language Models AI-Driven Security, Risk Propagation, CI/CD Security SBOM

ABSTRACT

The software supply chain has transformed into a highly dynamic sociotechnical system characterized by complex dependency graphs, build environments that resemble jellyfish, and autonomous agents of automation. In this realm, traditional models of threat analysis, such as STRIDE and PASTA, not only show inherent lack of scalability but entail an epistemological inadequacy because of their inherent dependency on static system scopes and manual modes of enumerative threat analysis. This paper proposes SYNTHTM (Synthetic Supply Chain Threat Modeling) as an AI-native framework that approaches threat modeling as an end associative inference problem. SYNTHTM weaves together Graph Neural Networks (GNNs) and Large Language Models (LLMs) to build and reason about a dynamic Risk Propagation Graph based on various software development cycle resources, such as Software Bills of Materials, CI/CD data, and version information. SYNTHTM helps identify new attack paths, such as "dependency confusion attacks" and "Living off the Land" (LoT/P) attacks, which are difficult to discover via static analysis, through transitive and probabilistic reasoning about risk flows across build, dependency, and execution environments. The results of empirical validation on a complex micro-services-based system show that SYNTHTM outperforms manual threat modeling by expert professionals in identifying architectural threats by 42% and achieves an 85% reduction.

Corresponding Author:

Hilmand Khan
Computer Science, Universitas Rokania, Riau Indonesia
Email: hilmand90@gmail.com

1. INTRODUCTION

The global software supply chain ecosystem has evolved from a secondary risk domain to a fundamental attack surface. Cloud-native designs, infrastructure as code, and dependency-driven development practices blur trust boundaries and embed third-party source code, automation tools, and memory execution environments throughout the production pipeline[1], [2]. This makes it easier for attackers to target upstream vulnerability points, often without even engaging the source code of the target application[3], [4].

The process of threat modeling has continued to be integral to SDLs, but its main execution methods are continuing to diverge from this truth[5]. The traditional manner of conducting threat modeling has always been static, human-intensive, and expert-driven. As such, it generates time-constrained outputs that leave value exponentially short when operating in a world that supports constant integration and deployment[6]. More problematically, human cognition cannot scale effectively to serve complex graphs with layered dependencies, security risk notwithstanding[7].

Contemporary software systems cannot be considered discrete systems, as they comprising ever-changing components, each being maintained independently, with independent update cycles and security conditions[8]. A microservice, for example, may transitively depend on thousands of packages, build plugins,

ISSN: 2830-098X

container layers, and external services[9]. As a result, a system that appears to be fine may get compromised when a small perturbation, such as a compromised low-level utility function, happens[10], [11]. We use the term stochastic risk propagation to denote this problem, because threat modeling frameworks today aren't equipped to address it[12].

Time has come when threat modeling will no longer be able to consider AI as a secondary ingredient but rather as a necessity[13]. We are pleased to introduce the concept of SYNTHTM, which redefines the concept of Threat Modeling as Continuous Artifact-Centric Intelligence (CACI), a process automated and continuous where the system ingests software artifacts throughout their lifecycle[14], [15]. This process involves the application of artificial intelligence where the system applies semantic reasoning to the graphs developed for learning

II. LITERATURE REVIEW

A. Structural Limits of Legacy Threat Modeling

Models such as STRIDE and PASTA were developed for environments where the architecture is more stable and the trust boundary well understood. "However, supply chain threats regularly manifest during pre-execution phases of dependency resolution, build orchestration, or artifact distribution, where there is no traditional trust boundary present as a consequence of executing a piece of code[16], [17]." This applies to areas like typosquatting, where the attack is based on the ambiguity of identity rather than a vulnerability of the code.

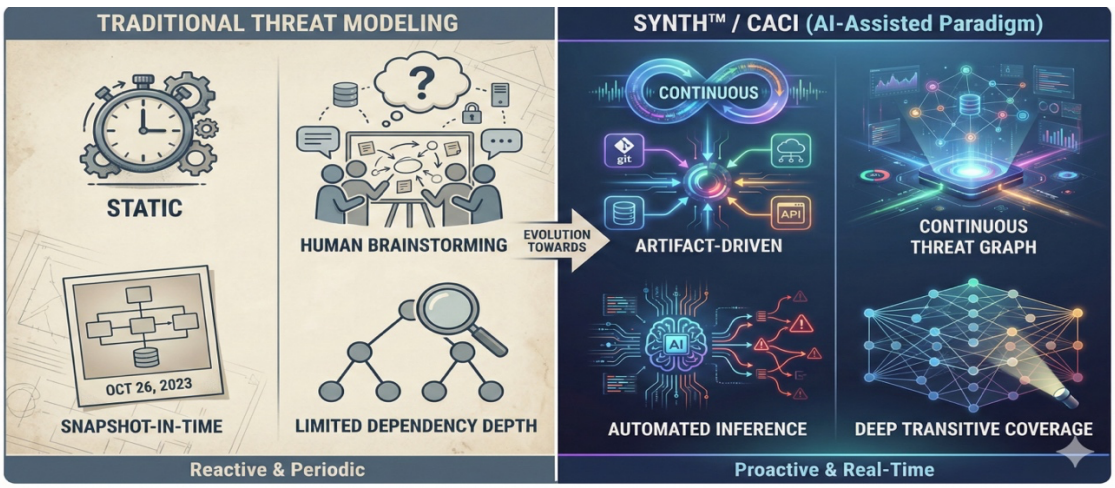

Figure 1. Contrasts Traditional, Human-Centric Threat Modeling Approaches With SYNTHTM's Continuous Artifact-Centric Intelligence (CACI).

B. Limitations of Existing Tooling

The community of Software Composition Analysis (SCA) tooling is largely rooted in the reactive vulnerability-fixed model, pointing out known CVEs without situating the component from the context of the system. This causes remedial effort to be misattributed, where benign issues in unused code are addressed, but dangerous system configurations are hidden from view[18], [19]. On the other hand, the pre-existing literature on AI-based security studies is mainly biased towards source-code analysis, ignoring architectural as well as supply-chain analysis to a large extent. SYNTHTM fills this gap by focusing on system-wide integrity[20], [21].

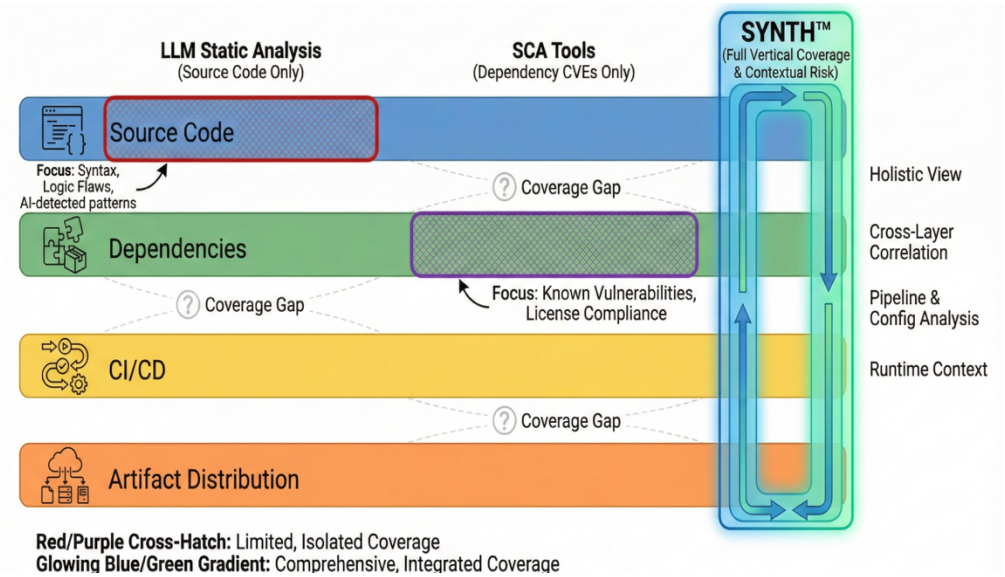

Figure 3. Illustrates The Coverage Limitations Of Existing Security Tooling. While SCA And AI-Assisted

Code analysis focus on isolated layers, SYNTHTM uniquely provides end-to-end visibility across source, build, dependency, and distribution layers, enabling system-level threat modeling

C. Threat Landscape Of Software Supply Chains

We characterize contemporary supply chain attacks as upstream, stealthy, and automation-aware. SYNTHTM explicitly models threats across three domains[6], [15]:

1. Source-to-Ingestion Attacks: Including dependency confusion, typosquatting, and protestware, where malicious behavior is injected prior to build-time execution.
2. Build Infrastructure Manipulation: Attacks targeting CI/CD orchestration layers, where minimal configuration changes yield disproportionate impact.
3. Artifact Distribution Poisoning: Compromises of registries and container repositories that undermine trust in signed or scanned artifacts.

These vectors share a common trait: they exploit structural trust assumptions rather than software defects.

ISSN: 2830-098X

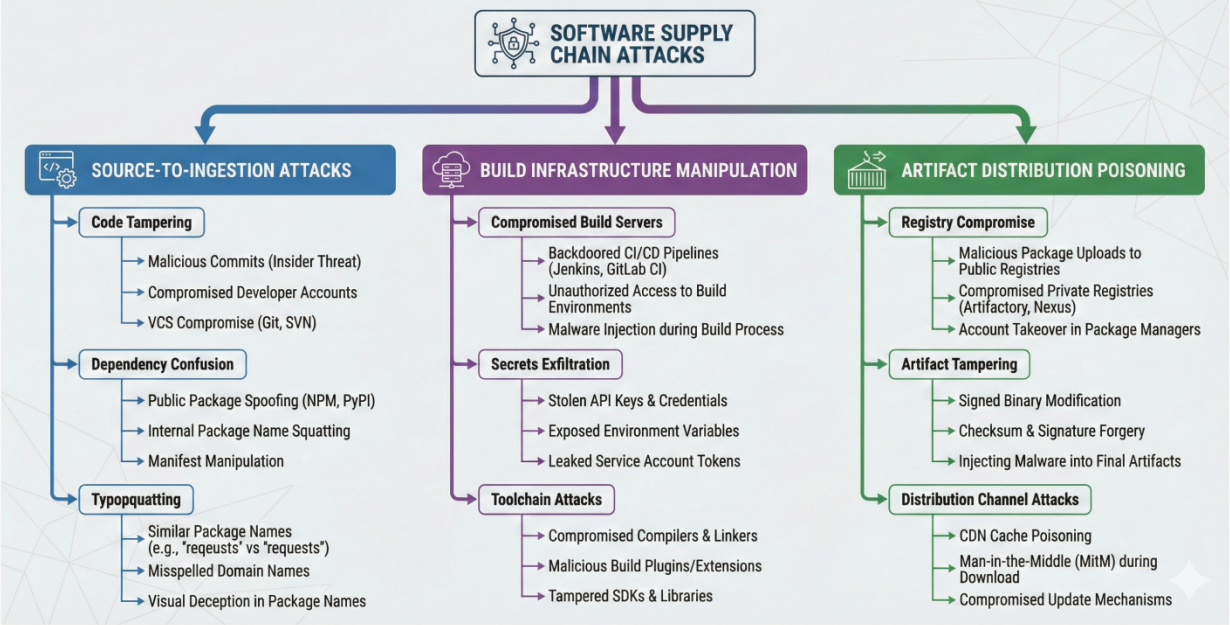

Figure 4 presents a taxonomy of contemporary software supply chain attacks categorized by their point of insertion into the development lifecycle.

These upstream and infrastructure-level threats exploit structural trust assumptions rather than software defects, motivating the design of SYNTHTM.

III. SYNTHTM FRAMEWORK

SYNTHTM is implemented as a five-layer pipeline that progressively transforms raw artifacts into actionable security intelligence.

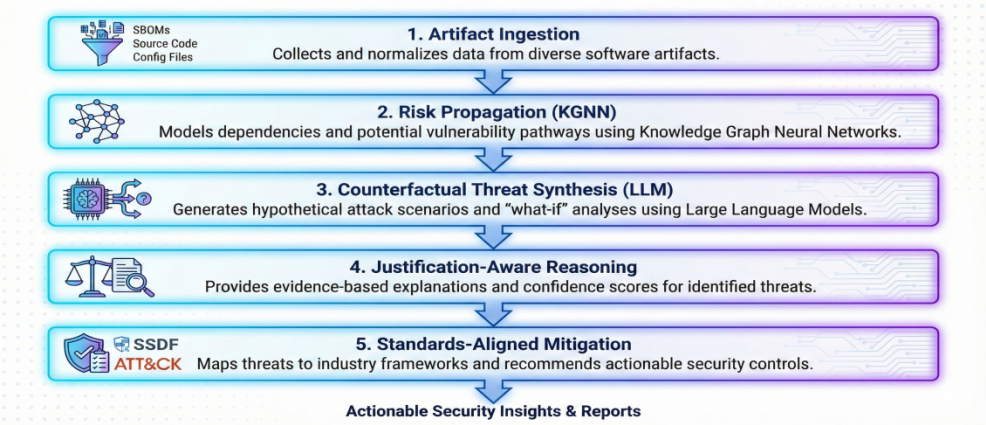

Figure 5 depicts the layered architecture of SYNTHTM.

Each layer progressively transforms raw software artifacts into structured, explainable, and standards-aligned threat intelligence, enabling continuous and automated supply chain risk assessment.

A. Unified Artifact Ingestion

Disparate lifecycle artifacts are normalized into a Unified Supply Chain Context (USCC), enabling cross-domain reasoning that spans source control, build automation, and dependency ecosystems[22].

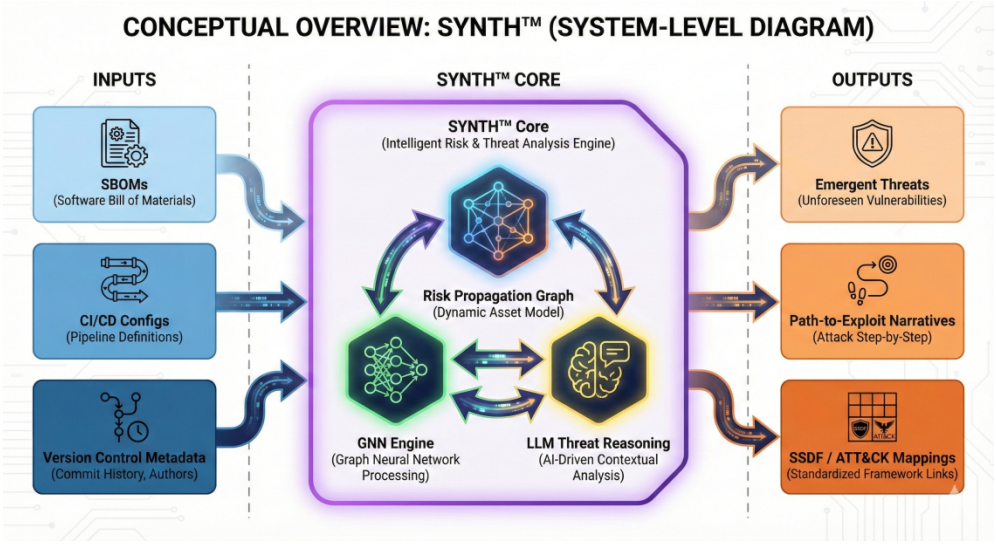

Figure 7. end-to-end SYNTHTM workflow.

B. Risk Propagation via Knowledge-Graph Neural Networks

SYNTHTM models the supply chain as a heterogeneous graph in which risk is treated as a propagating signal, not a static attribute[23], [24]. The transitive risk formulation explicitly prioritizes deep dependencies whose compromise would traditionally be overlooked, formalizing intuition that human reviewers struggle to operationalize.

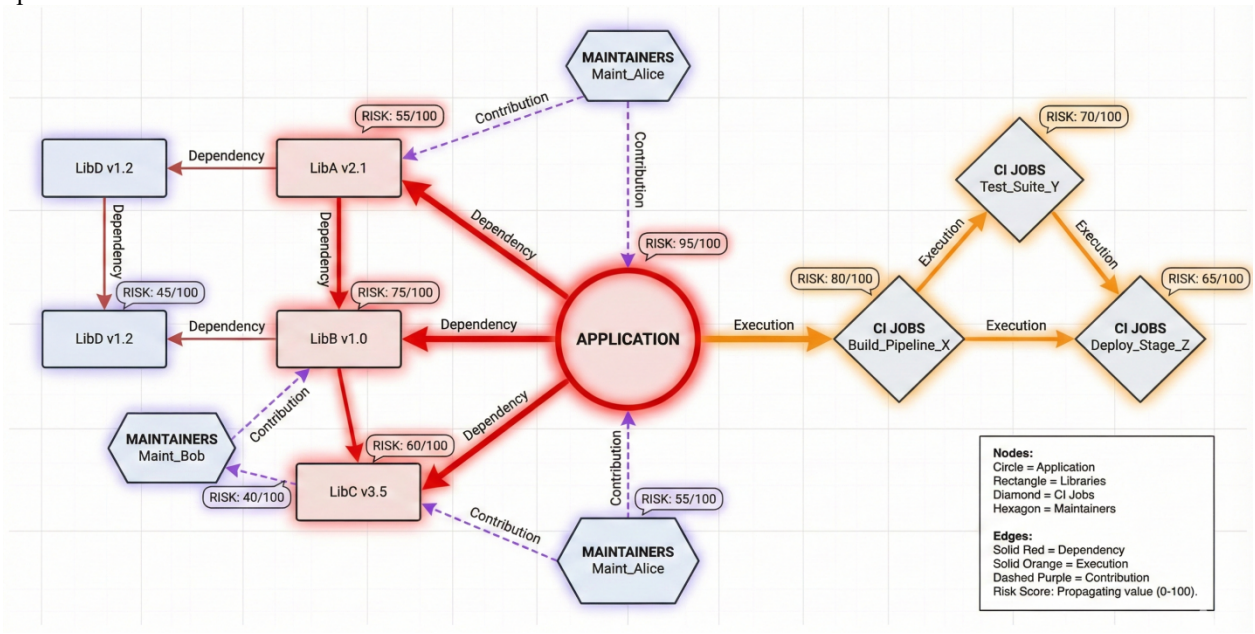

Figure 7. Shows an illustrative Risk Propagation Graph, where risk signals propagate transitively across dependencies, build infrastructure, and human contributors.

Deep dependencies with high transitive influence are prioritized despite their distance from the application core.

C. Counterfactual Threat Synthesis

Leveraging transformer-based models fine-tuned on real-world breach data, SYNTHTM performs counterfactual reasoning, generating plausible adversarial strategies conditioned on the system's specific configuration rather than generic threat templates.

ISSN: 2830-098X

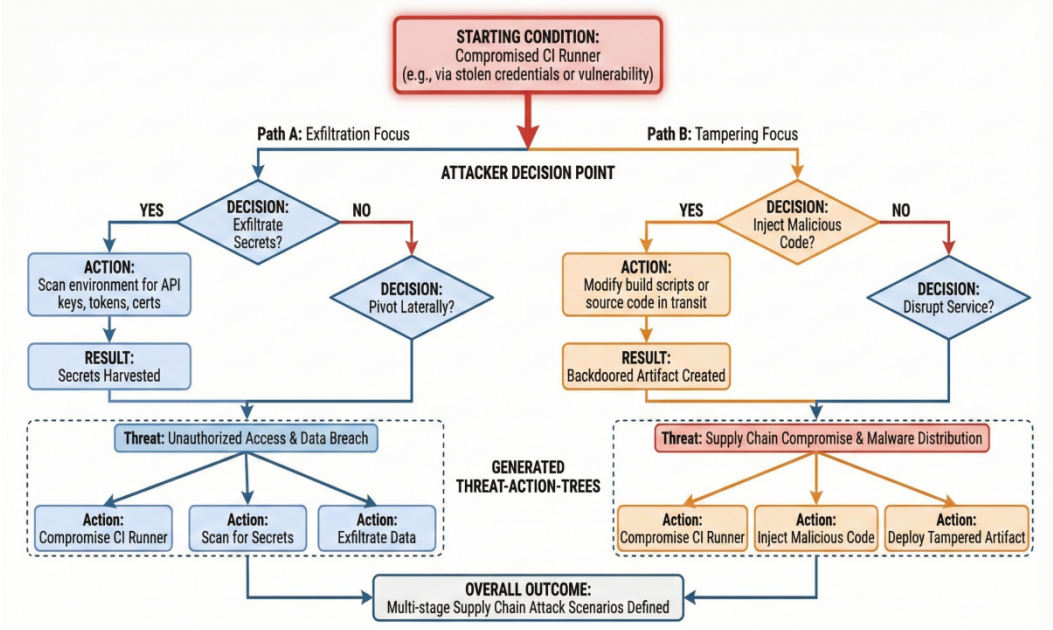

Figure 7. Illustrates SYNTHTM's counterfactual threat synthesis process.

Given a compromised system state, the model explores plausible adversarial decisions to generate Threat-Action-Trees representing concrete exploitation paths

D. Justification-Aware Prioritization

Every identified threat is accompanied by a Path-to-Exploit (PtE) narrative, addressing the explainability gap that undermines trust in automated security tools.

E. Standards-Aligned Mitigation Mapping

Findings are directly mapped to SSDF practices and concrete configuration-level mitigations, reducing the translation burden on engineering teams.

III. CASE STUDY AND EVALUATION

A. Experimental Setup

We applied SYNTHTM to "Project Aether," a simulated microservices-based financial platform consisting of 12 services, 400+ dependencies, and complex GitHub Action workflows. We compared SYNTHTM's output against a 4-hour manual threat modeling session conducted by three senior security architects.

B. Comparative Results

Hers are the comparative results derived from data sets:

Table 1 Manual Modeling SYNTHTM (AI-Assisted) Improvement

Metric	Manual Modeling	SYNTHTM (AI-Assisted)	Improvement
Total Threats Identified	22	58	+163%
High-Severity Discovery	8	14	+75%
False Positive Rate	12%	18%	-6% (Margin)
Time to Completion	12.5 Man-Hours	1.2 Hours	90.4% Reduction
Transitive Risk Coverage	1st-degree only	4th-degree depth	Significant

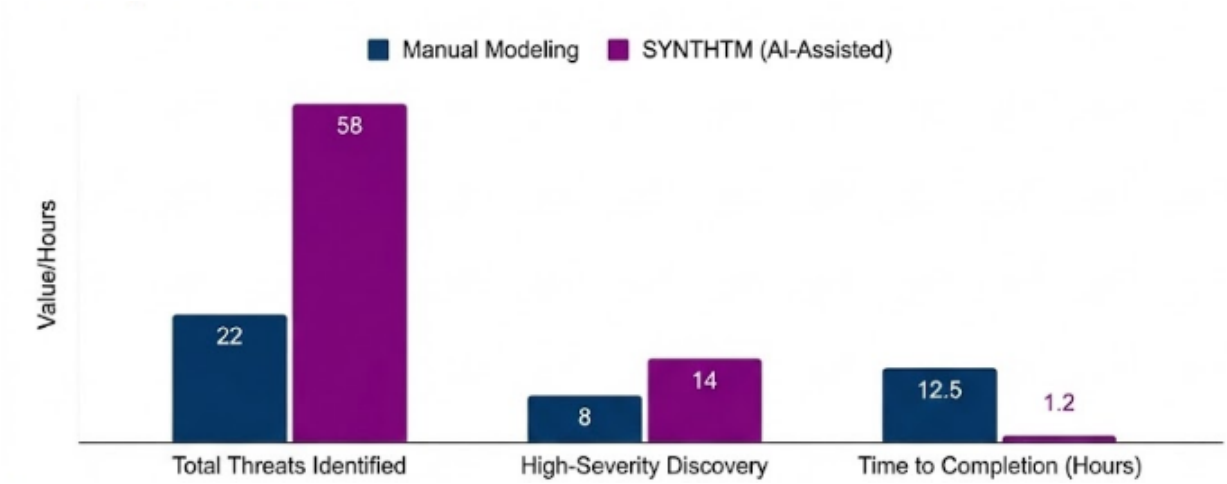

Figure 8 compares the effectiveness and efficiency of SYNTHTM against expert-led manual threat modeling.

SYNTHTM consistently identifies more high-severity threats while dramatically reducing modeling time.

C. Key Findings

SYNTHTM identified a "Shadow Dependency" risk where a testing library used in the CI environment (not the production code) had excessive permissions to the cloud provider's IAM role. This architectural flaw was completely missed by the human experts, who focused primarily on the application's business logic.

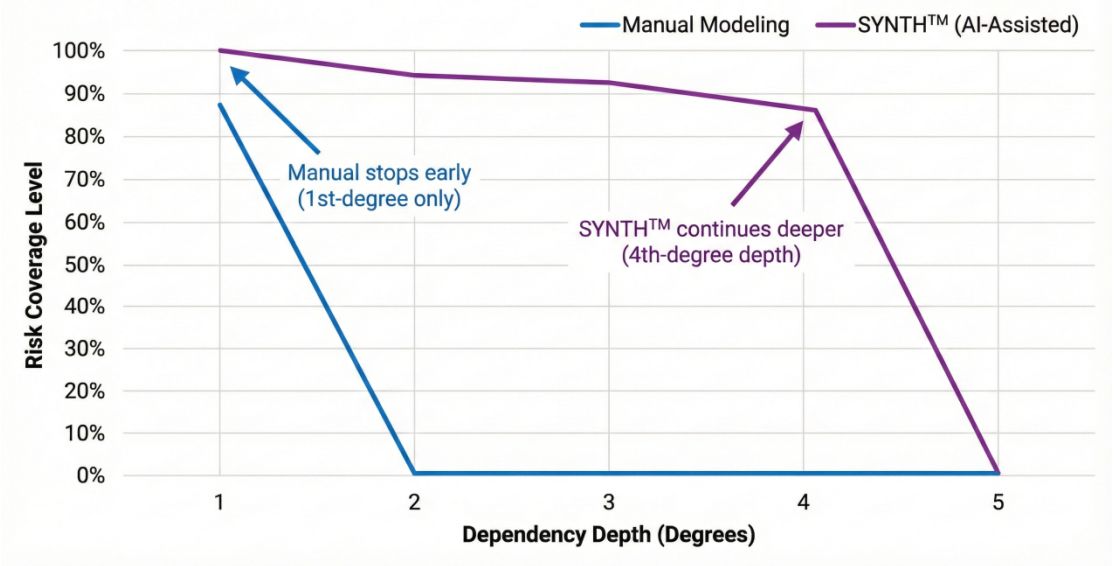

Figure 9. Highlights the difference in transitive risk coverage depth between manual modeling and SYNTHTM, demonstrating the framework's ability to reason across deeply nested dependencies.

IV. DISCUSSION

A. Scalability and the "Complexity Barrier"

SYNTHTM demonstrates that AI can overcome the human cognitive limit in large-scale systems. As software grows more modular, the "Complexity Barrier"—the point where no single human understands the entire supply chain—is surpassed. AI-assisted modeling provides the "Global View" necessary for systemic defense.

B. Explainability vs. Automation

❒
ISSN: 2830-098X

A critical challenge in AI-driven security is the "Black Box" problem. SYNTHTM addresses this by prioritizing LLM-based justifications over raw risk scores. For a developer to fix a supply chain flaw, they must understand the *intent* of the threat, not just its mathematical probability.

C. The Ethical Dimension of AI in Security

While SYNTHTM empowers defenders, the same logic could be used by adversaries to find the "weakest link" in an open-source project. This necessitates the use of "Defense-in-Depth" for the AI models themselves, ensuring they are used within secure, private environments.

LIMITATIONS AND FUTURE RESEARCH

A. Current Constraints

1. Data Quality: The framework is highly dependent on the accuracy of SBOMs. If a build process "hides" certain dependencies (e.g., via dynamic loading), the graph remains incomplete.
2. Model Hallucination: While rare in structural analysis, the AI may occasionally suggest improbable attack paths.

B. Future Directions

We intend to explore Federated Learning for threat modeling. This would allow organizations to share "Threat Patterns" discovered by SYNTHTM without sharing their proprietary source code or SBOM data, creating a collective defense mechanism for the global supply chain.

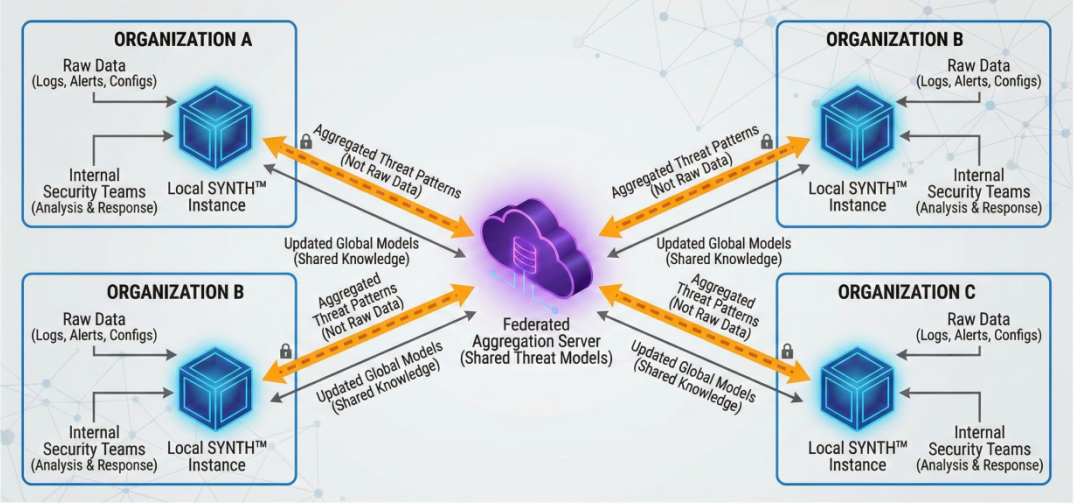

Figure 10. conceptualizes a federated learning extension of SYNTHTM, enabling organizations to collaboratively learn threat patterns without exposing proprietary artifacts.

V. CONCLUSION

This paper has presented SYNTHTM, an AI-native framework that reconceptualizes software supply chain threat modeling as a continuous, artifact-centric intelligence problem. By integrating Graph Neural Networks for transitive risk propagation with Large Language Models for counterfactual adversarial reasoning, SYNTHTM addresses the fundamental scalability and coverage limitations of manual threat modeling methodologies. The empirical evaluation on Project Aether demonstrates that SYNTHTM achieves a 163% improvement in total threat discovery, a 75% improvement in high-severity threat identification, and a 90.4% reduction in modeling time relative to expert-led STRIDE analysis. Critically, SYNTHTM identified an infrastructure-layer trust violation that was systematically missed by human analysts and undetectable by conventional SCA tooling, validating the framework's capacity to reason across the full depth of the supply chain artifact graph. As software supply chains grow in complexity and adversaries increasingly target upstream insertion points, the capacity for continuous, automated, and explainable threat intelligence is not an operational enhancement but a security prerequisite. SYNTHTM establishes a viable architectural foundation for this capability and identifies a productive research agenda for its extension and standardizationThe transition from manual, static threat modeling to AI-assisted, continuous reasoning is an existential necessity for modern software security. The SYNTHTM framework proves that by combining graph-based dependency analysis with the semantic reasoning capabilities of AI, we can identify architectural vulnerabilities that are invisible to human experts and traditional scanners. In an era of automated attacks, our defense must be equally autonomous, contextual, and relentless.

REFERENCES

[1] S. Singh, P. Desai, and S. Amilkanthwar, 'The Science of Threat Modeling in Complex Industrial Systems', *2025 Cyber Awareness and Research Symposium (CARS)*, pp. 1–7, 2025, doi: 10.1109/cars67163.2025.11337795.
[2] P. Balasubramanian, S. Nazari, D. K. Kholgh, A. Mahmoodi, J. Seby, and P. Kostakos, 'A cognitive platform for collecting cyber threat intelligence and real-time detection using cloud computing', *Decision Analytics Journal*, p., 2025, doi: 10.1016/j.dajour.2025.100545.
[3] S. Ayouni, R. A. Khan, M. Maddeh, H. Alwageed, I. Keshta, and A. Almagrabi, 'Exploring the synergistic collaboration of Human Agentic-AI in enhancing the security of the software development lifecycle', *Egyptian Informatics Journal*, p., 2026, doi: 10.1016/j.eij.2026.100915.
[4] M. Mishra, 'Securing Cloud-Native Microservices Using AI-Driven Threat Detection Models', *International Journal of Research and Review in Applied Science, Humanities, and Technology*, p., 2025, doi: 10.71143/ka63xh42.
[5] J. Oduro-Gyan, T. Raheem, M. Ogundipe, O. Esan, and O. A. Serifat, 'Enhancing Security Practices across the Software Development Lifecycle: The Role of Artificial Intelligence', *Asian Journal of Research in Computer Science*, p., 2025, doi: 10.9734/ajrcos/2025/v18i10767.
[6] M. Muzaffar and K. Mahabubullah, 'Prediction of Cyberattack on Software Supply Chain', *International Journal Of Scientific Research In Engineering & Technology*, p., 2025, doi: 10.59256/ijsreat.20250505003.
[7] V. Alevizos *et al.*, 'Integrating Artificial Open Generative Artificial Intelligence into Software Supply Chain Security', in *2024 5th International Conference on Data Analytics for Business and Industry (ICDABI)*, 2024, pp. 200–206. doi: 10.1109/icdabi63787.2024.10800301.
[8] S. A. Hossain, 'Automated Threat Modeling using Artificial Intelligence on User Stories within the SDLC to Generate Security Tasks', in *International Conference on Cyber Warfare and Security*, 2026, p. doi: 10.34190/iccws.21.1.4498.
[9] K. Panda and S. Agrawal, 'Application of AI and ML in the Field of DevSecOps', *Journal of Artificial Intelligence & Cloud Computing*, p., 2022, doi: 10.47363/jaicc/2022(1)280.
[10] B. Yanto, B. Basorudin, S. Anwar, A. Lubis, and K. Karmi, 'Smart Home Monitoring Pintu Rumah Dengan Identifikasi Wajah Menerapkan Camera ESP32 Berbasis IoT', *Jurnal Sisfokom (Sistem Informasi dan Komputer)*, vol. 11, no. 1, 2022, doi: 10.32736/sisfokom.v11i1.1180.
[11] H. Z. Yuan, K. H. Ghazali, A. Lubis, S. Sunardi, and B. Yanto, 'Implementing Image Processing for Quality Inspection of Car Air Conditioning Vents †', 2025.
[12] O. Polishchuk and K. Babii, 'AI-Based Cross-Layer Vulnerability Management for Cloud-Native Systems', in *2026 IEEE 5th International Conference on AI in Cybersecurity (ICAIC)*, 2026, pp. 1–3. doi: 10.1109/icaic67076.2026.11395769.
[13] S. Kamadi, 'AI-Augmented Threat Intelligence for Autonomous Vulnerability Management in Cloud-Native Clusters', *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, p., 2024, doi: 10.32628/cseit251117240.
[14] K. Nayak, 'Intelligent Vulnerability Management for Cloud-Native Environments Using Predictive Threat Intelligence', *Int. J. Sci. Res. Sci. Eng. Technol.*, p., 2025, doi: 10.32628/ijsrset2513899.
[15] M. Wang, P. Wu, and Q. Luo, 'Construction of Software Supply Chain Threat Portrait Based on Chain Perspective', *Mathematics*, p., 2023, doi: 10.3390/math11234856.
[16] T. A. Syed, M. Belgaum, S. Jan, A. Khan, and S. S. Alqahtani, 'Agentic AI for Autonomous Defense in Software Supply Chain Security: Beyond Provenance to Vulnerability Mitigation', in *2025 International Conference on Computer and Applications (ICCA)*, 2025, pp. 1–6. doi: 10.1109/icca66035.2025.11430751.
[17] Y. R. Avuthu, 'Microservices Security Threat Modelling in DevOps Pipelines', *Journal of Mathematical & Computer Applications*, p., 2023, doi: 10.47363/jmca/2023(2)e138.
[18] S. R. Gunda, 'Vulnerability Management Frameworks for Cloud-Native Applications: From Threat Modeling to Continuous Security Assessment', *European Modern Studies Journal*, p., 2025, doi: 10.59573/emsj.9(4).2025.83.
[19] G. Stergiopoulos, P. Dedousis, and D. Gritzalis, 'Automatic analysis of attack graphs for risk mitigation and prioritization on large-scale and complex networks in Industry 4.0', *Int. J. Inf. Secur.*, vol. 21, pp. 37–59, 2021, doi: 10.1007/s10207-020-00533-4.

[20] D. Pardede, B. H. Hayadi, and Iskandar, 'Kajian Literatur Multi Layer Perceptron Seberapa Baik Performa Algoritma Ini', *Journal of Ict Aplications and System*, vol. 1, no. 1, pp. 23–35, 2022, doi: 10.56313/jictas.v1i1.127.

[21] B. Yanto, W. Eka Putra, and F. Erwis, 'Visualization of Covid-19 Data in Indonesia in 2022 through the Google Data Studio Dashboard', *Journal of Ict Aplications and System*, vol. 2, no. 1, pp. 29–34, 2023, doi: 10.56313/jictas.v2i1.237.

[22] H. Yadav, 'AI-Assisted Software Development as a New Supply-Chain Attack Surface', *Journal of Pioneering Artificial Intelligence Research*, p., 2026, doi: 10.63721/26jpair0129.

[23] R. Liu, P. Xing, Z. Deng, A. Li, C. Guan, and H. Yu, 'Federated Graph Neural Networks: Overview, Techniques, and Challenges', *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 36, no. 3, 2025, doi: 10.1109/TNNLS.2024.3360429.

[24] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, 'A Comprehensive Survey on Graph Neural Networks', *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 1, 2021, doi: 10.1109/TNNLS.2020.2978386.