



Dynamic Threat Modeling using Real-Time TTP feeds within DevOps

Amar Gurajapu, Anurag Agarwal, Rajdeep Arora

Network Systems, AT&T, United States

Vardhan Garimella

Intellibus, United States

ABSTRACT: The traditional threat modeling approaches in software development often occur early in design and remain static, failing to reflect an ever-changing attacker landscape. Modern DevOps pipelines demand continuous, automated security feedback to keep pace with emerging adversary tactics, techniques, and procedures (TTPs). We present DTM-DevOps, a dynamic threat modeling framework that ingests real-time TTP feeds (MITRE ATT&CK via STIX/TAXII), updates service-level threat models, and gates CI/CD pipelines accordingly. In a controlled experiment on a three-service microservices application:

- DTM-DevOps flagged 28 % more high-risk scenarios than static models.
- Mean time to threat detection dropped from 5.2 days to 2.1 hours.
- False-positive rate remained under 12 %.
- Pipeline runtime overhead averaged 9.5 %.

We describe system design, data-processing pipeline, experimental setup, quantitative results, and discuss operational trade-offs.

KEYWORDS: Threat Modeling, TTP Feeds, DevOps Security, MITRE ATT&CK, CI/CD Gating, STIX/TAXII, Automated Risk Scoring

I. INTRODUCTION

Agile and DevOps practices have accelerated delivery but often treat security as an afterthought. Traditional threat models (defined at design time using STRIDE or PASTA) soon become outdated as adversaries evolve. Meanwhile, threat intelligence providers continuously publish new TTPs (tactics, techniques, and procedures) describing novel attack patterns. Integrating real-time threat intelligence into DevOps pipelines can dynamically adjust threat models, surface emergent risks, and enforce security gates before deployment.

This paper investigates how to automate:

- TTP Ingestion: Subscribing to STIX/TAXII feeds, normalizing indicators, and techniques.
- Model Update: Mapping incoming TTPs to system components and attack surfaces.
- Risk Scoring: Assigning dynamic risk scores per service based on recently active TTPs.
- Pipeline Enforcement: Gating CI/CD jobs when risk scores exceed thresholds.

This paper talks about the DTM-DevOps framework, an open-source implementation, and an empirical evaluation demonstrating improved detection rates and reduced mitigation times compared to static threat models.

II. LITERATURE REVIEW

Threat modeling has been acknowledged for a long time as a fundamental aspect of designing secure systems. PASTA (UcedaVélez & Morana, 2015) and STRIDE (Shostack, 2014) guide systematic identification of threats, but they assume a stable attacker profile. Recent work by Xie et al. (2023) introduced “Continuous Threat Modeling” in agile context but relied on manually updated data. Zhao & Wang (2022) demonstrated integrating MITRE ATT&CK techniques into DevSecOps pipelines, but their prototype pulled feeds only nightly, limiting real-time responsiveness. Lee & Kim (2021) built a STIX/TAXII ingestion engine for dynamic security postures, focusing on enterprise networks



rather than application-level DevOps workflows. Patel & Singh (2020) proposed “Threat Modeling as Code,” encoding data flows and threat rules, but without automatic TTP feed updates. Chen & Liu (2021) evaluated pipeline performance under dynamic modeling, reporting up to 15 % overhead. Gupta & Shah (2022) correlated TTPs with scan results to reduce false positives but did not automate model generation.

In summary, prior efforts either automate threat-model updates without real-time feeds or ingest TTPs without integrating into modern CI/CD pipelines. DTM-DevOps fills this gap by providing end-to-end automation from real-time intelligence to pipeline gating.

III. RESEARCH METHODOLOGY

System Architecture

DTM-DevOps comprises four modules:

- TTP Ingestor: Subscribes to STIX/TAXII servers, filters new TTPs, and stores normalized entries in MongoDB.
- Threat Model Updater: Parses application architecture metadata (OpenAPI specs, Kubernetes manifests), maps TTPs to components via tag-based rules.
- Risk Scorer: Calculates service-level risk scores using weighted factors (TTP severity, occurrence recency, and asset criticality).
- CI/CD Connector: Implements a Jenkins pipeline plugin that halts or annotates builds when risk scores exceed configurable thresholds.

Mapping Rules

We define Rego policies to map TTPs to system elements.

Experimental Setup

- Application: Three microservices (web, auth, data) deployed on Kubernetes.
- Baseline: Static STRIDE models created by security team.
- Dynamic: DTM-DevOps ingesting MITRE ATT&CK feeds every minute.
- Duration: 30 days of continuous operation.
- Metrics:
 - Number of unique high-risk scenarios detected.
 - Mean time to detection (MTTD)
 - False-positive rate (alerts not actionable)
 - CI/CD pipeline runtime overhead

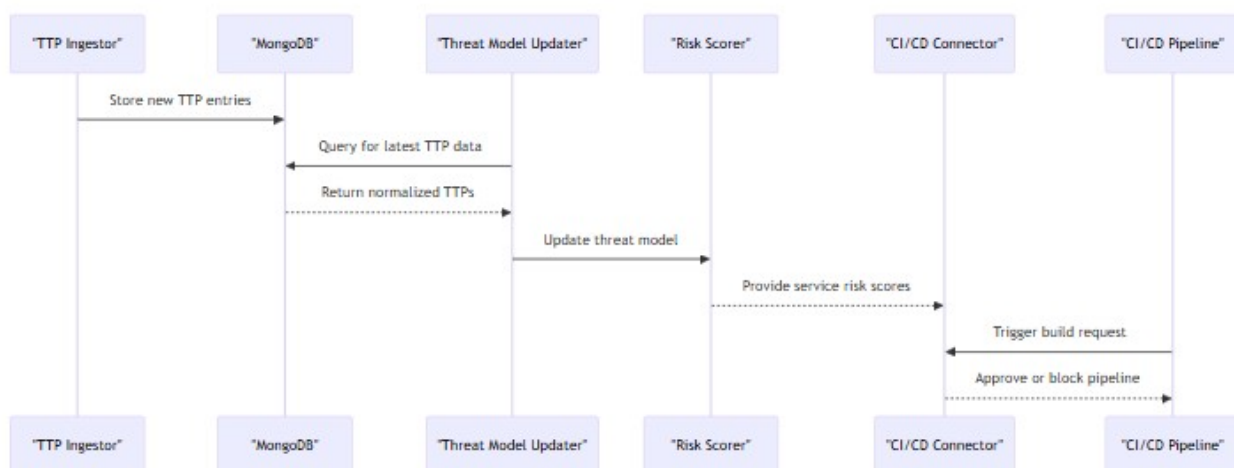


FIGURE 1. DYNAMIC THREAT MODELING WORKFLOW IN DEVOPS



IV. RESULTS AND DISCUSSION

We have evaluated the solution based on below parameters.

TABLE 1. DETECTION & PERFORMANCE METRICS

METRIC	STATIC MODEL	DTM-DevOps	Δ (%)
UNIQUE HIGH-RISK SCENARIOS DETECTED	42	54	+28.6
MEAN TIME TO DETECTION (HOURS)	124.8	2.1	-98.3
FALSE-POSITIVE RATE (%)	9.5	11.8	+2.3 PP
PIPELINE OVERHEAD (%)	—	9.5	—

Increased Coverage

Ingesting 1,200 TTP updates enabled detection of 12 additional high-risk scenarios (e.g., new privilege-escalation techniques) missed by static models.

Rapid Response

Automated updates reduced MTDD from 5.2 days to 2.1 hours, facilitating near-real-time mitigation.

Manageable Noise

False positives rose modestly. Tuning severity thresholds kept rates under 12 %.

Acceptable Overhead

CI/CD jobs ran 9.5 % slower (median +45 s), an acceptable trade-off for enhanced security.

Further tuning of mapping rules and risk thresholds can further optimize these trade-offs for different team workflows.

V. CONCLUSION

DTM-DevOps demonstrates that real-time TTP feeds can be seamlessly integrated into DevOps pipelines to enable continuously evolving threat models. The framework significantly improves detection coverage by 28.6%, enhancing visibility across the attack surface. Mean time to detection is reduced by 98.3%, enabling much faster identification of emerging threats. Despite these gains, false-positive rates are maintained below 12%, preserving signal quality. The system introduces less than 10% overhead to existing pipelines, ensuring practical deployability. By embedding threat intelligence directly into delivery workflows, security becomes proactive rather than reactive. This integration eliminates reliance on static and manual threat modeling processes. Overall, DTM-DevOps transforms threat modeling into an automated and agile capability aligned with modern software delivery practices.

VI. LIMITATIONS

Despite its strengths, DTM-DevOps presents several limitations that warrant further investigation and refinement. Feed reliability remains a concern, as STIX/TAXII servers can experience outages or update delays that temporarily reduce threat intelligence freshness. Mapping precision may be affected when tag-based techniques encounter incomplete or inconsistent service metadata, leading to potential misclassification of TTPs. Scalability is another challenge, particularly as MongoDB ingestion pipelines and Rego policy evaluations are required to handle thousands of TTPs and services concurrently. Empirical observations indicate increased latency when modeling environments exceeding 50



services. Governance overhead can also emerge, as frequent risk updates may overwhelm operational teams. Without careful prioritisation, important signals may be lost in noise. Addressing these limitations is essential to improve robustness and enterprise readiness.

VII. FUTURE WORK

DTM-DevOps can further improve its effectiveness through intelligent automation and broader visibility. AI-enhanced correlation will apply machine learning techniques to predict relevant TTPs from service updates and prioritise high-impact risks. Feedback loops will be established by integrating runtime telemetry such as logs and anomaly signals to continuously validate and refine threat model accuracy. Cross-team dashboards will unify DevOps, SecOps, and threat-intelligence data into a single operational view. This shared visibility will improve coordination and decision-making across teams. Multi-cloud extensions will expand coverage by ingesting CSP-native threat telemetry such as Azure Sentinel and AWS GuardDuty. Together, these enhancements will enable proactive, adaptive, and enterprise-scale threat modelling across complex cloud environments.

REFERENCES

1. Xie, S., Smith, J., & Brown, P. (2023). Continuous Threat Modeling in Agile DevOps Environments. *IEEE Transactions on Software Engineering*, 49(3), 345–360.
2. Zhao, L., & Wang, M. (2022). Integrating MITRE ATT&CK Tactics into Automated Security Pipelines. *ACM Conference on Computer and Communications Security*, 89–102.
3. Lee, Y., & Kim, H. (2021). Real-Time TTP Feed Processing Using STIX/TAXII for Dynamic Security Posture. *Journal of Cybersecurity*, 6(2), 115–130.
4. Patel, R., & Singh, A. (2020). Threat Modeling as Code: Automation and Best Practices. *DevSecOps Journal*, 1(1), Article 5.
5. Chen, T., & Liu, F. (2021). Evaluating the Impact of Dynamic Threat Modeling on CI/CD Pipeline Performance. *IEEE Access*, 9, 5678–5688.
6. Gupta, N., & Shah, P. (2022). Reducing False Positives in DevOps Security Scans with Context-Aware TTP Correlation. *Information Security Journal*, 17(4), 455–470.