Toward Performance-Based Detection of Attacks in Microservice Systems

Andrea Janes
ajanes@unibz.it
Free University of Bozen-Bolzano
Bolzano, Italy

Diaeddin Rimawi
diaeddin.rimawi@unibz.it
Free University of Bozen-Bolzano
Bolzano, Italy

Refat Tahseen Othman
refatta.othman@unibz.it
Free University of Bozen-Bolzano
Bolzano, Italy

Alberto Avritzer
beto@esulabsolutions.com
eSulabSolutions
Princeton, New Jersey, USA

Raffaela Mirandola
raffaela.mirandola@kit.edu
Karlsruher Institut für Technologie
Karlsruhe, Germany

Barbara Russo
barbara.russo@unibz.it
Free University of Bozen-Bolzano
Bolzano, Italy

Abstract

This *poster* paper links performance engineering and cybersecurity by presenting an architecture for performance-based attack detection in microservice systems. We define a workflow based on service-level runtime behavior rather than network signatures or logs. The architecture integrates workload generation, attack injection, and continuous collection of CPU, memory, filesystem activity, and response-time metrics. In future work, we will evaluate this architecture on benchmarking applications that reflect realistic architectures reported by practitioners. The goal is to assess whether runtime performance behavior can support service-level attack detection.

CCS Concepts

• **Software and its engineering → Software performance**; • **Security and privacy → Intrusion detection systems**; **Denial-of-service attacks**; **Distributed systems security**.

Keywords

Performance signatures, Anomaly detection, Microservices, Security monitoring, Intrusion detection

ACM Reference Format:
Andrea Janes, Diaeddin Rimawi, Refat Tahseen Othman, Alberto Avritzer, Raffaela Mirandola, and Barbara Russo. 2026. Toward Performance-Based Detection of Attacks in Microservice Systems. In *Companion of the 17th ACM/SPEC International Conference on Performance Engineering (ICPE Companion '26), May 04–08, 2026, Florence, Italy.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3777911.3799732

1 Introduction

Performance engineering and cybersecurity are increasingly intertwined in modern software systems. While performance engineering focuses on analyzing resource usage and responsiveness, security attacks often manifest as deviations in system performance behavior. Microservice-based applications continuously monitor runtime performance metrics to aid operational tasks, providing a detailed view of service behavior under different conditions [1, 2].

Attacks targeting microservices can induce observable changes in performance metrics, such as increased resource consumption or abnormal response times. Despite this, performance monitoring is typically used for reliability and efficiency analysis rather than for identifying security threats. As a result, the potential of performance data as an indicator of attack conditions remains underexplored [6].

In microservice-based systems, attacks may affect individual services or propagate through shared infrastructure, leading to complex performance effects. Existing security monitoring approaches often rely on explicit indicators such as suspicious network packets, authentication failures, or known attack patterns. However, attacks that are novel, obfuscated, or resource-oriented may evade these mechanisms while still causing performance degradation.

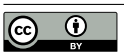
In this *poster* paper, we present an architecture for performance-based attack detection in microservice architectures. We define how service-level performance behavior can be monitored under benign and adversarial conditions. The objective of this work is to assess whether deviations in runtime performance behavior can be used to detect attacks at the service level, without relying on network signatures or application logs. In the next phase, we will begin evaluation with DeathStarBench [4], as depicted in Figure 1, and then continue with additional benchmark applications that reflect realistic architectures reported by practitioners.

Rather than proposing a new intrusion detection system, this study defines an architecture for investigating the potential of performance metrics as indicators of attack conditions. This poster paper is guided by the following research question: *can performance-based indicators alone expose attacks in microservice architectures?*

Our contribution is architectural. Specifically, we define:

- the system and tooling pipeline for workload generation, attack execution, and telemetry collection;
- the performance-metric data flow used for service-level observation;
- a reproducible workflow to compare benign and adversarial executions.

In contrast to prior work on performance signatures [2], we focus on an architecture for containerized microservice deployments and explicit service-level attack mapping, with evaluation planned as future work.



ICPE Companion '26, Florence, Italy

ACM ISBN 979-8-4007-2326-1/2026/05
https://doi.org/10.1145/3777911.3799732

2 Experimental Setup and Methodology

We define a performance-based attack-analysis architecture for containerized microservice systems. In future evaluation, we will first instantiate this architecture on DeathStarBench and then extend it to additional benchmark applications, including suites with multiple interacting services and realistic end-to-end functionalities common in practice.

A representative workload is generated by PPTAM[1] using Locust[2], which emulates concurrent users interacting with the system through the frontend service. The workload configuration is selected to ensure stable and responsive system behavior under benign conditions and is kept unchanged during adversarial executions to allow direct comparison between normal and attack scenarios. During execution, service-level performance metrics are continuously collected using container-level monitoring tools. The monitored metrics include CPU utilization, memory consumption, filesystem input/output activity, and request response times, providing a fine-grained view of runtime service behavior.

The architecture defines a workflow with a benign execution phase to establish a performance baseline, followed by adversarial phases in which targeted and combined attack scenarios are applied to selected services under the same workload conditions. The initial planned attack classes include service-level denial-of-service and resource-exhaustion patterns, with both isolated and combined injections, to stress specific services through controlled workload and attack placement.

Figure 1 provides an overview of the setup: PPTAM performs load testing using Locust and measures response times. In a two-machine setup, attack scenarios are injected from the Driver/Attacker machine according to the architecture design. On the server machine, the software under test is executed (e.g., DeathStarBench applications [4], TrainTicket[3], or generated benchmarking applications [5]) and its resource consumption is measured using cAdvisor[4]; the Driver reads CPU/memory/IO metrics exposed by cAdvisor.

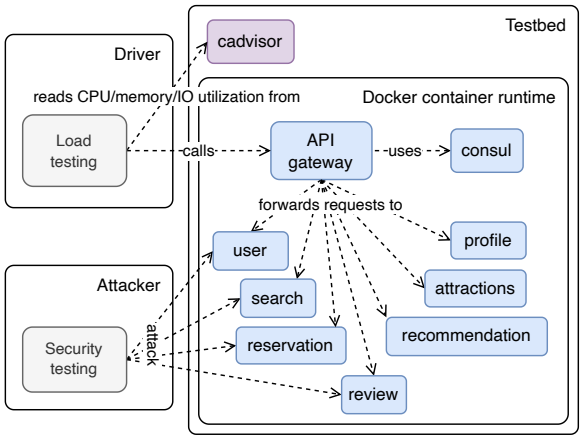


Figure 1: C4 container diagram [3] depicting the Testbed containing the software under test, its main containers, as well as two external systems: Driver, which performs load testing and reads CPU/memory/IO metrics exposed by cAdvisor, and Attacker, which performs different attacks on components of the SUT. The current setup is illustrated with a DeathStarBench-based SUT.

3 Discussion and Future Work

This work emphasizes the close relationship between performance engineering and cybersecurity in microservice-based systems and contributes an architecture to study that relationship. From a performance engineering perspective, the architecture leverages existing monitoring infrastructures already used for reliability and capacity analysis, requiring no specialized security instrumentation.

As future work, we will execute the planned attack scenarios against individual services in selected benchmark applications and instantiate the full workflow end to end. For each service, the collected performance behavior under attack will be compared against a benign baseline to identify deviations induced by adversarial activity. We also plan to study how attack-induced performance deviations propagate across services and to evaluate how this architecture can support practical, performance-aware security monitoring.

Open questions remain regarding which additional attacks should be evaluated beyond this initial set, e.g., low rate or stealthy attacks that subtly degrade performance, cache related attacks, or frontend resource exhaustion. This, to better understand the limits of the system. It is also unclear which detection algorithms are most suitable, ranging from simple threshold or change point methods to more advanced anomaly detection or graph based approaches that exploit inter service dependencies. It also remains an open question which potential use cases this approach has, e.g., supporting site reliability engineers in early detection of abnormal behavior. We are also interested in understanding which microservice benchmarks and which workloads are particularly suited to the described approach. Finally, attacks with minimal performance impact, slow stealthy behavior, or logic level abuse may slip through this approach and need to be characterized.

[1] https://github.com/pptam/pptam-tool
[2] https://locust.io
[3] https://github.com/FudanSELab/train-ticket
[4] https://github.com/google/cadvisor

References

[1] Alberto Avritzer, Andre Bondi, Michael Grottke, Kishor S. Trivedi, and Elaine J. Weyuker. 2006. Performance Assurance via Software Rejuvenation: Monitoring, Statistics and Algorithms. In *Proceedings of the International Conference on Dependable Systems and Networks (DSN '06)*. IEEE Computer Society, USA, 10 pages.

[2] Alberto Avritzer, Rajanikanth Tanikella, Kiran James, Robert G. Cole, and Elaine Weyuker. 2010. Monitoring for security intrusion using performance signatures. In *Proceedings of the First Joint WOSP/SIPEW International Conference on Performance Engineering* (San Jose, California, USA) *(WOSP/SIPEW '10)*. Association for Computing Machinery, New York, NY, USA, 12 pages.

[3] Simon Brown. 2026. *The C4 Model: Visualizing Software Architecture.* O'Reilly Media. https://www.oreilly.com/library/view/the-c4-model/9798341660113/

[4] Yu Gan, Yanqi Zhang, Dailun Cheng, Christina Delimitrou, et al. 2019. An Open-Source Benchmark Suite for Microservices and Their Hardware-Software Implications for Cloud & Edge Systems. In *Proceedings of the Twenty-Fourth International Conference on Architectural Support for Programming Languages and Operating Systems* (Providence, RI, USA) *(ASPLOS '19)*. Association for Computing Machinery, New York, NY, USA, 16 pages.

[5] Yannik Lubas, Martin Straesser, André Bauer, and Samuel Kounev. 2025. Generating Executable Microservice Applications for Performance Benchmarking. In *Proceedings of the 16th ACM/SPEC International Conference on Performance Engineering* (Toronto ON, Canada) *(ICPE '25)*. Association for Computing Machinery, New York, NY, USA, 14 pages.

[6] S. S. Nasim et al. 2025. A Systematic Literature Review on Intrusion Detection Systems for Cloud Computing. *Discover Computing* 28, 107 (2025).