Performance Analysis of a Container Based Security Approach in 5G Networks

Alessandro Carrega
DITEN – University of Genoa
CNIT – S2N National Lab
alessandro.carrega@unige.it

Roberto Bruschi
DITEN – University of Genoa
CNIT – S2N National Lab
roberto.bruschi@unige.it

Musab M. W. MohamedAli
DITEN – University of Genoa
CNIT – S2N National Lab
musab.mohamedali@cnit.it

Ramin Rabbani
DITEN – University of Genoa
CNIT – S2N National Lab
ramin.rabbani@cnit.it

Agenda

- Introduction
- Motivation & Problem
- Research Objective
- Experimental Setup & Tools Overview
- Methodology
- Results & Comparative Analysis
- Conclusion

Introduction

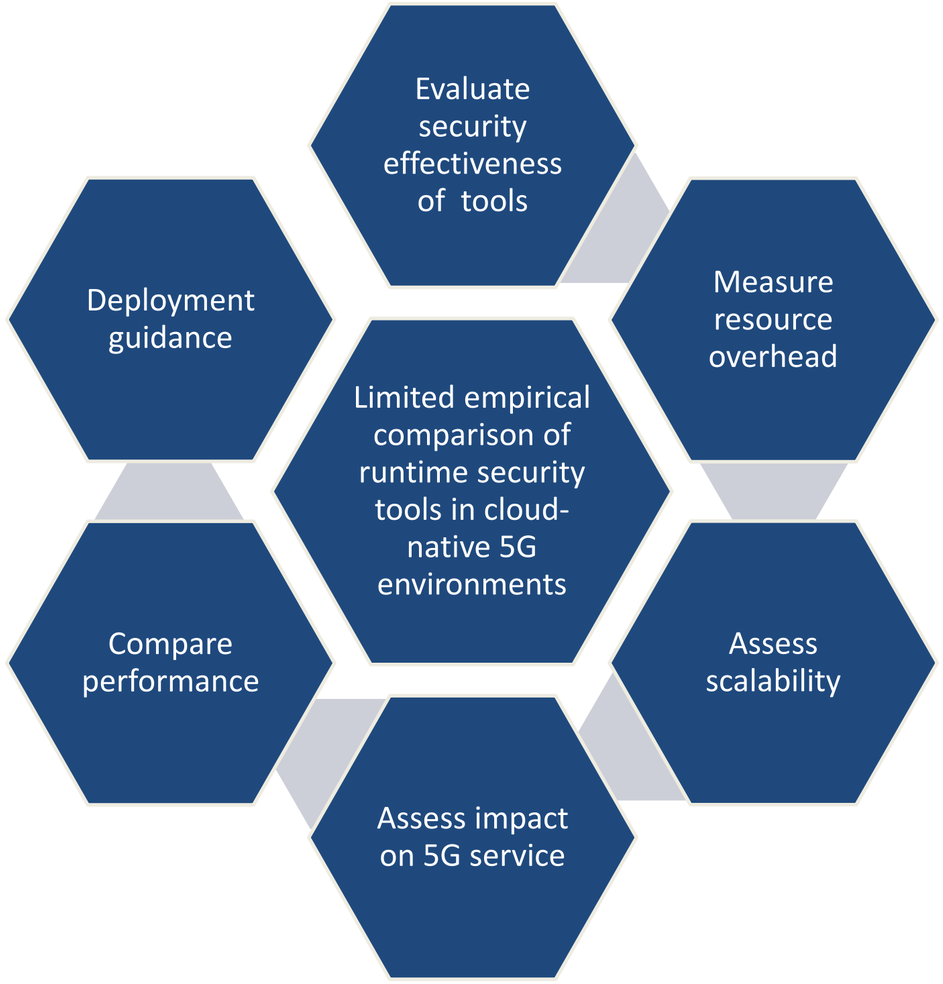
5G networks are transitioning to cloud-native, containerized architectures.

Kubernetes has become the standard for orchestrating 5G network functions.

Security gap: Despite widespread Kubernetes adoption for orchestrating 5G network functions.

Critical need: Balance security effectiveness with performance requirements.

Motivation & Problem



Research Objective

Evaluated two container-native security platforms in a 5G testbed

- Falco
- NeuVector

Evaluate performance overhead and monitoring effectiveness

- CPU usage
- Memory Consumption
- Scalability Under Event Load
- Detection Effectiveness

Identify the performance–security trade-off in cloud-native 5G deployments

- Lightweight Monitoring
- Comprehensive Security Coverage

Experimental Setup & Tools Overview

Testbed

- 3-node Kubernetes cluster.
- 1 control-plane node: 8 vCPUs, 8 GB RAM.
- 2 worker nodes: 6 vCPUs, 8 GB, 16 GB RAM.
- OS: Ubuntu 24.04.3 LTS.
- Container runtime: containerd 1.7.27.

Falco

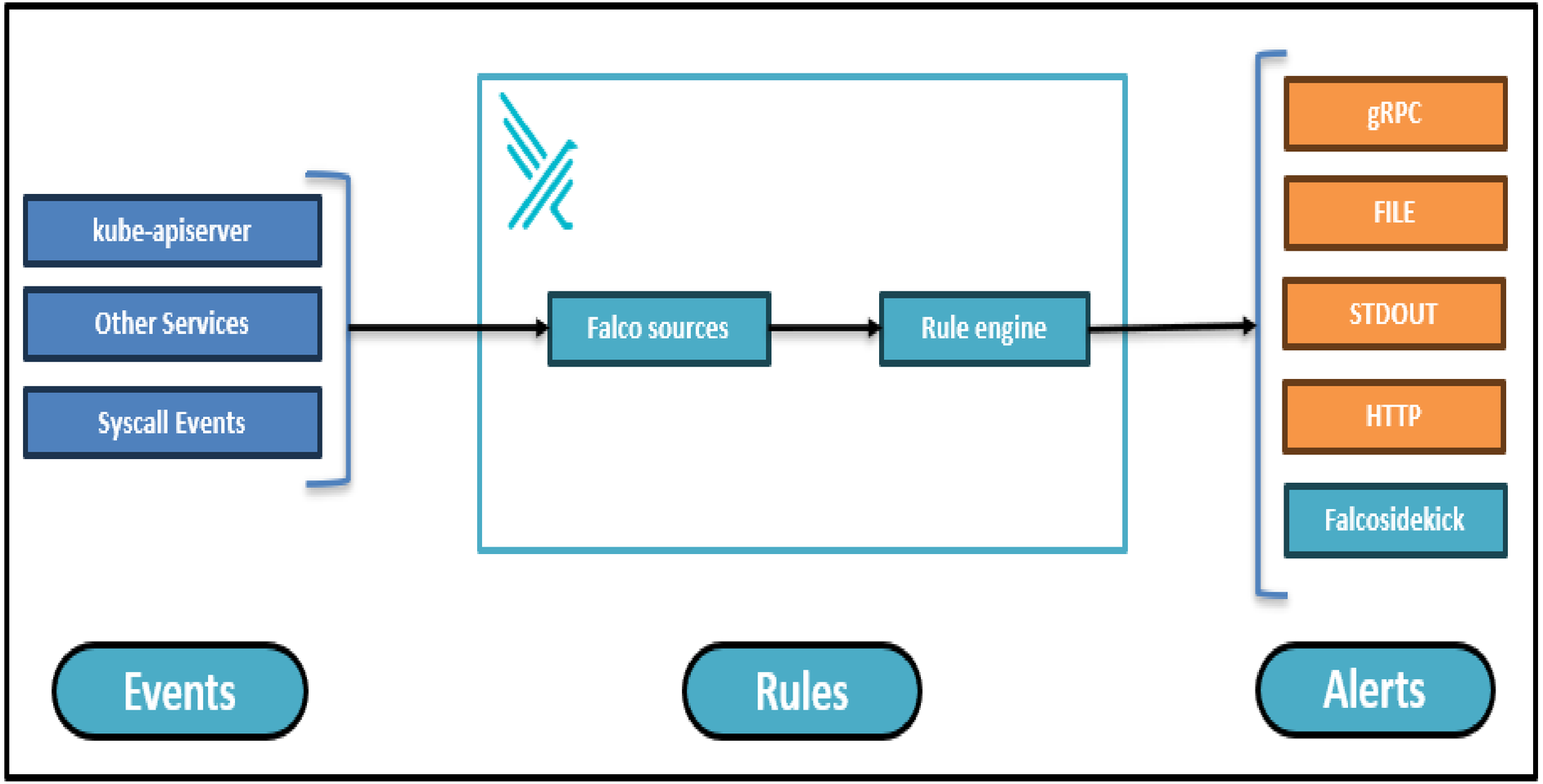
- Lightweight runtime security.
- Monitors kernel system calls.
- Rule-based detection.
- eBPF-based monitoring.

NeuVector

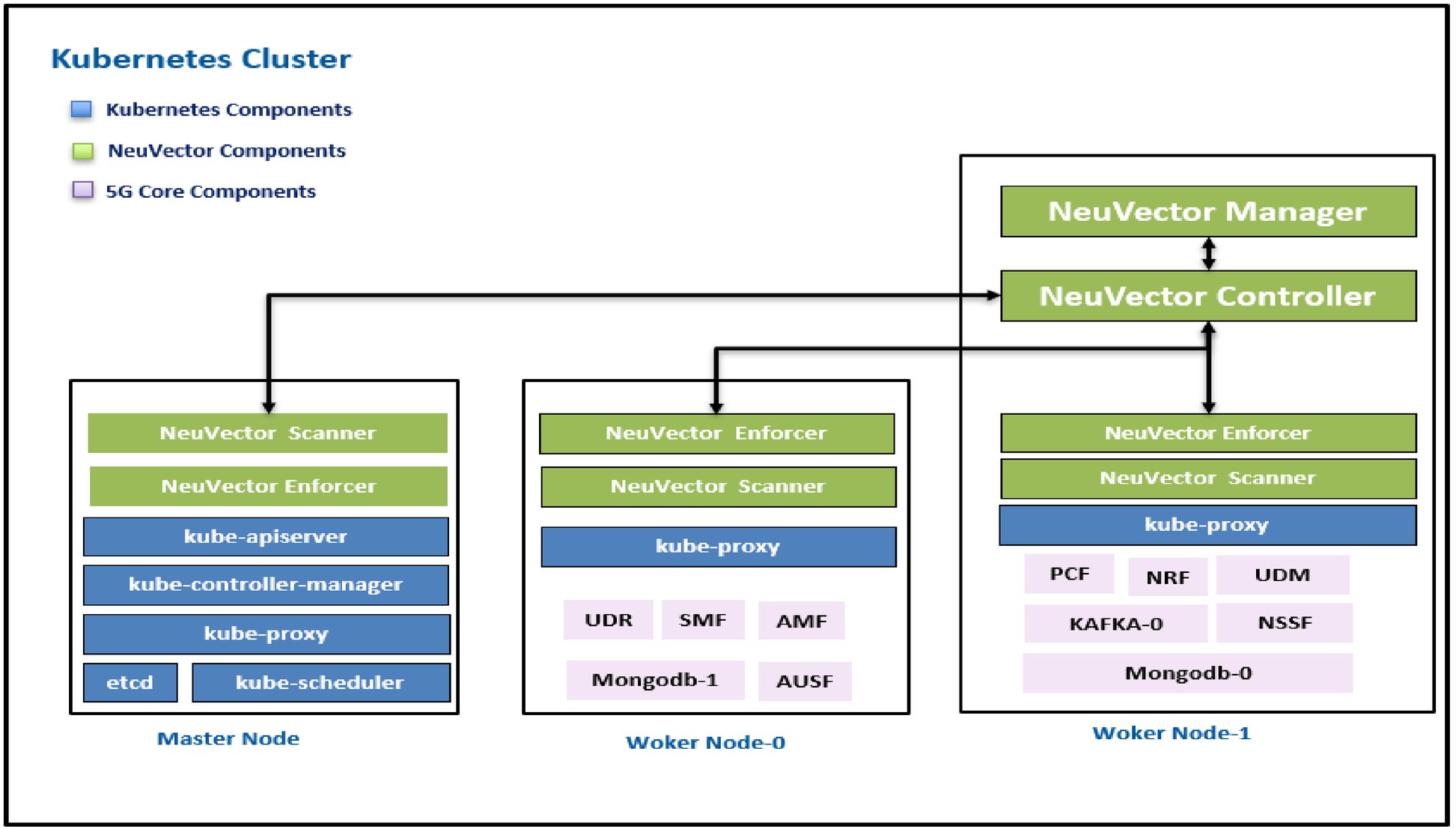
- Full-stack container security.
- Monitors processes, network traffic, and policy compliance.
- Supports enforcement and broader protection.

➢ ***Both platforms were deployed as DaemonSet-based node-level monitoring components.***

Falco


kube-apiserver
Other Services
Syscall Events
Falco sources
Rule engine
gRPC
FILE
STDOUT
HTTP
Falcosidekick
Events
Rules
Alerts

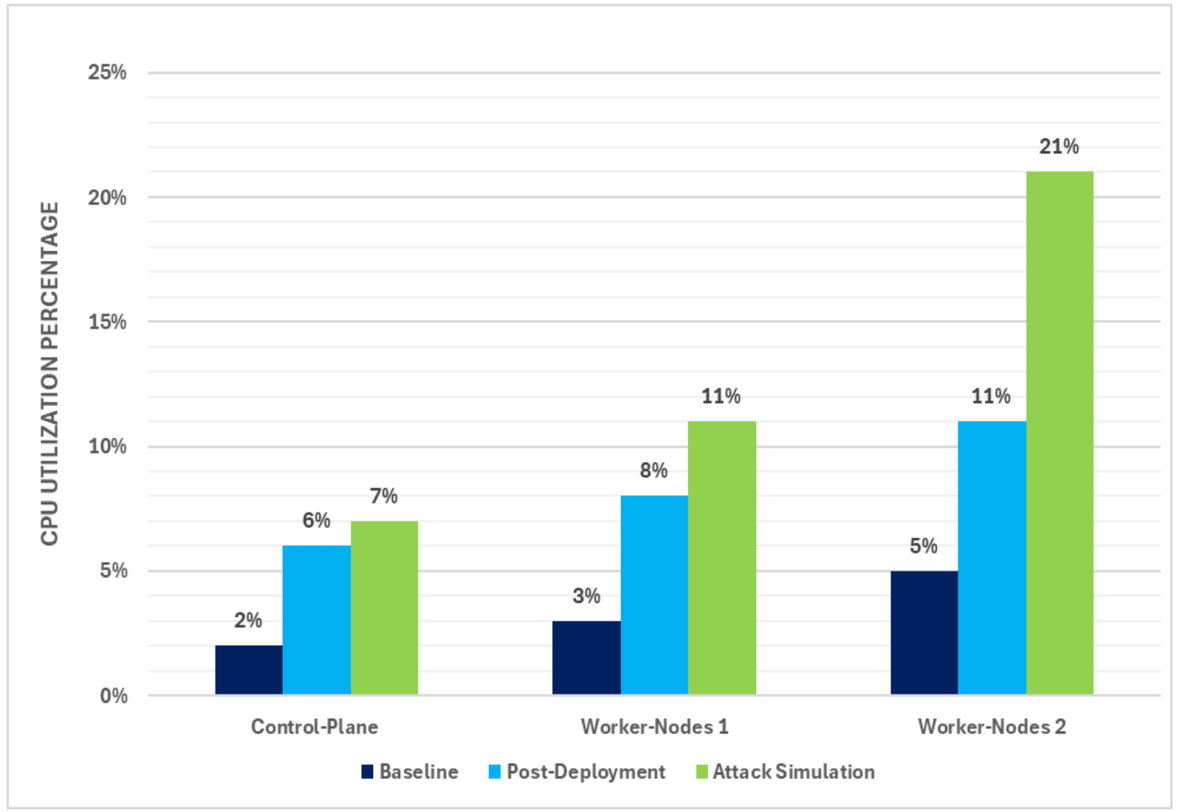
NeuVector


Kubernetes Cluster
Kubernetes Components
NeuVector Components
5G Core Components
NeuVector Manager
NeuVector Controller
NeuVector Scanner
NeuVector Enforcer
kube-apiserver
kube-controller-manager
kube-proxy
etcd
kube-scheduler
Master Node
NeuVector Enforcer
NeuVector Scanner
kube-proxy
UDR
SMF
AMF
Mongodb-1
AUSF
Woker Node-0
NeuVector Enforcer
NeuVector Scanner
kube-proxy
PCF
NRF
UDM
KAFKA-0
NSSF
Mongodb-0
Woker Node-1

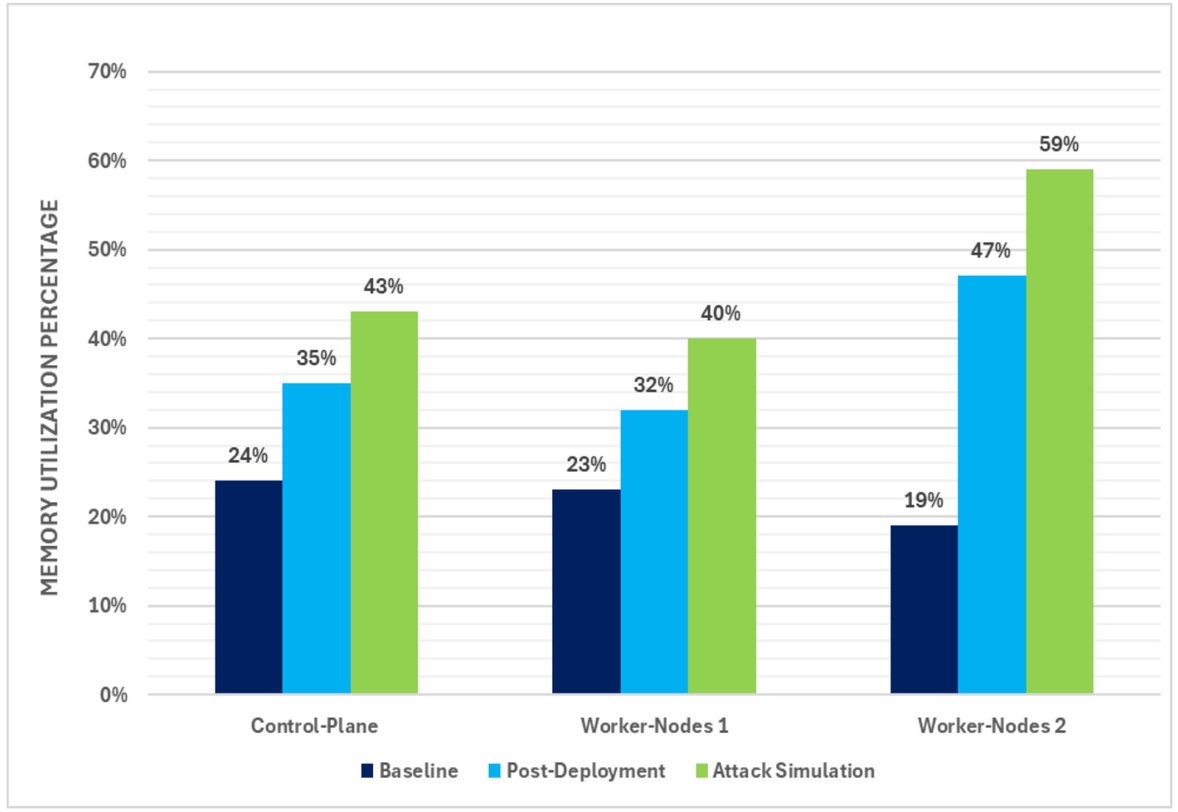
Methodology

Three Evaluation Phases	• Baseline • Post-Deployment • Attack Simulation
Metrics Collected	• CPU utilization • memory usage • system load averages
Double-Baseline Strategy	• Double-baseline used to reduce variability from Kubernetes scheduling and resource allocation
Event Generator	• Simulates attack scenarios and security-relevant activities • Covers container runtime and network behaviors • 400 attack scenarios in a 10-minute window
Ground-Truth Validation	• Ground truth established by correlating event generator logs with alerts • Matching based on timestamp, node identifier, and event type

Results – NeuVector Performance



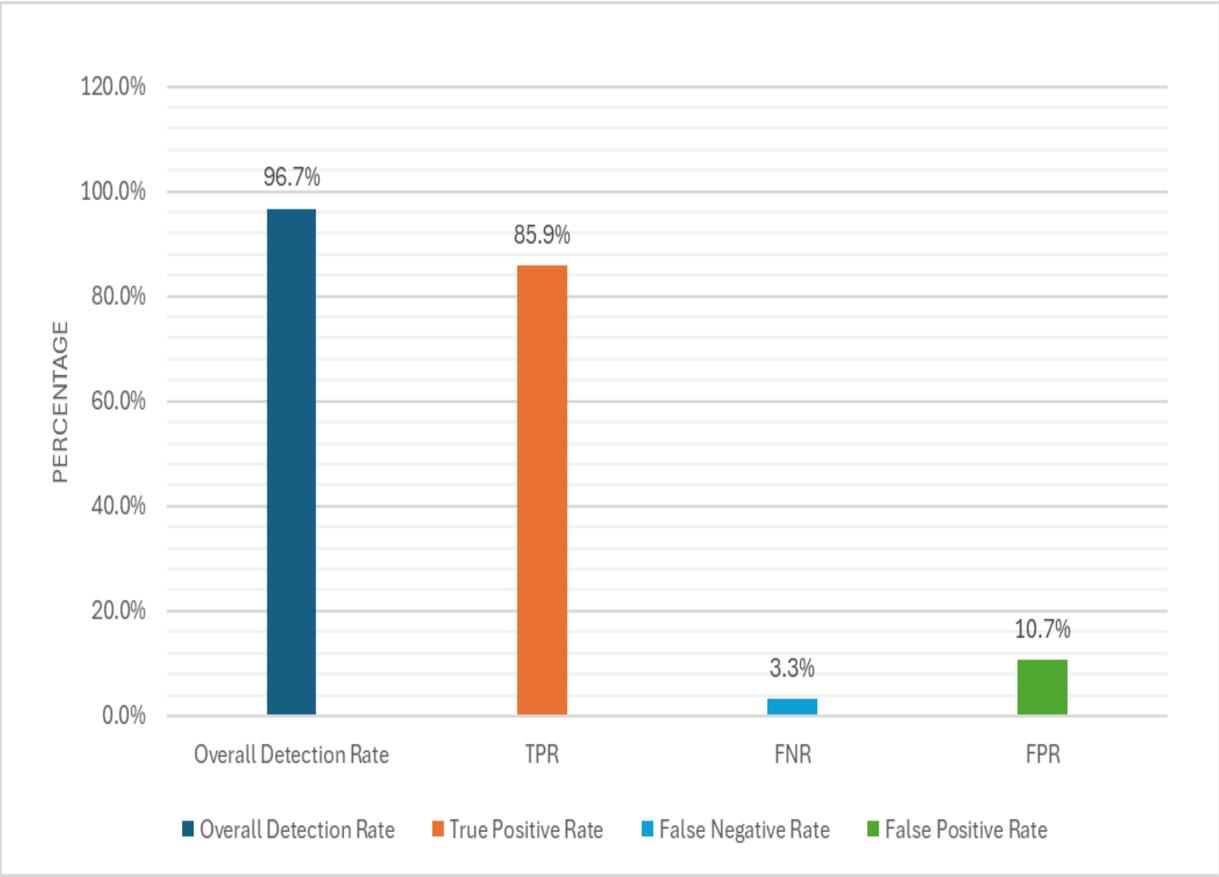
Absolut CPU Utilization Baseline, Post-Deployment and Attack Simulation



Absolut Memory Utilization Baseline, Post-Deployment and Attack Simulation

- **NeuVector shows moderate baseline overhead and stronger resource growth under attack load, especially on Worker Node 2.**

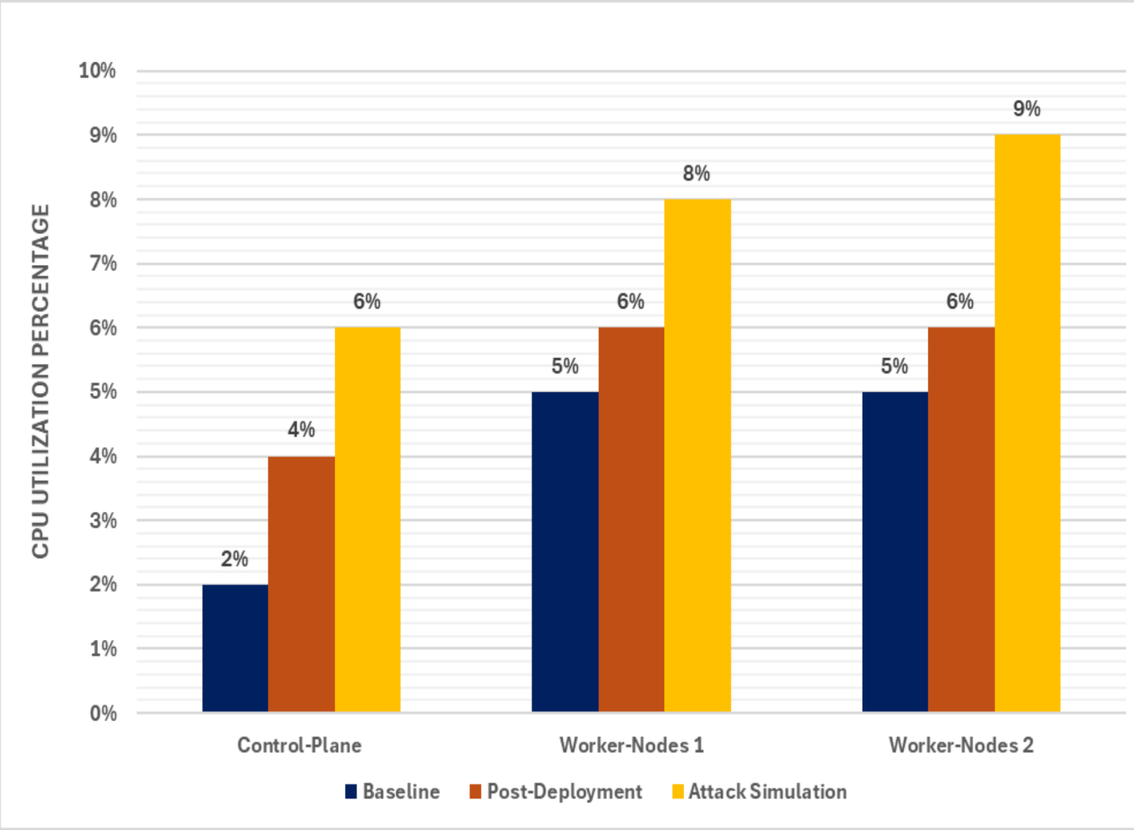
Results – NeuVector Performance



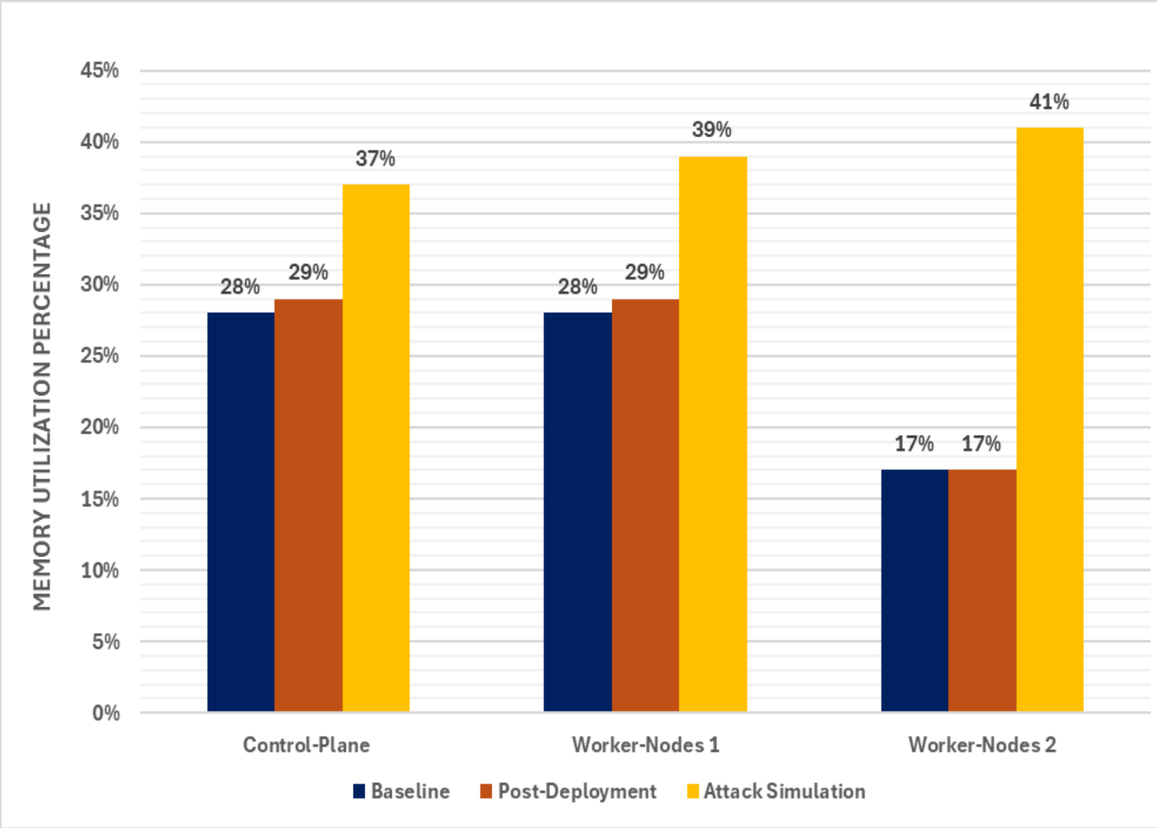
Security Effectiveness Comparison

Metric	Value	Count	Severity
False Positive Rate	10.7%	43/400	Warning Events
False Negative Rate	3.3%	13/400	Missed
True Positive Rate	85.9%	344/400	Blocked Threats
Overall Detection Rate	96.7%	387/400	Detected
Response Time	< 1 second	Real-time	Real-time

Results – Falco Performance



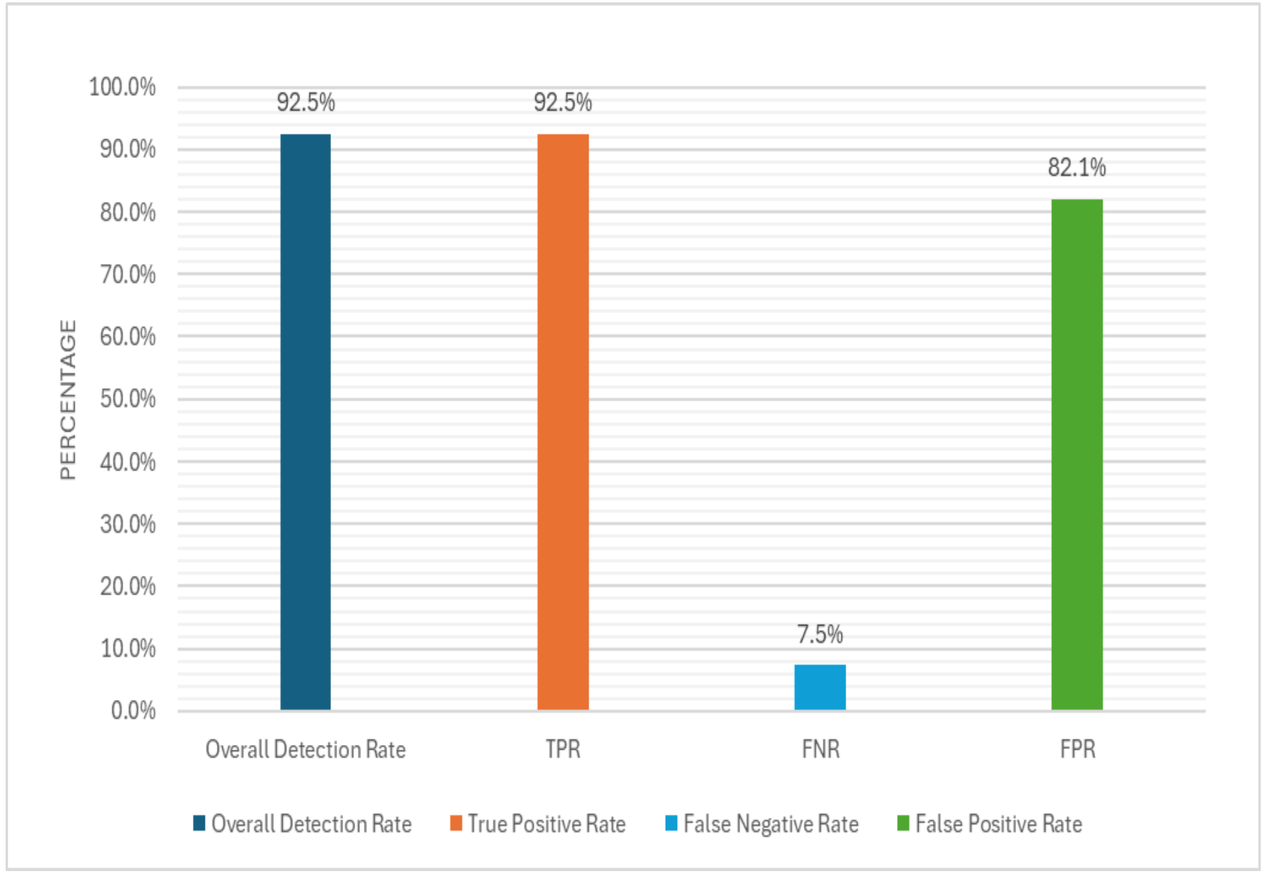
Absolut CPU Utilization Baseline, Post-Deployment and Attack Simulation



Absolut Memory Utilization Baseline, Post-Deployment and Attack Simulation

- **Falco maintains a lighter footprint across nodes, with a smaller post-deployment resource increase than NeuVector.**

Results – Falco Performance



Security Effectiveness Comparison

Metric	Value	Count	Severity
False Positive Rate	82.1%	1,700/2,100	Notice
False Negative Rate	7.5%	30/400	Missed
True Positive Rate	92.5%	370/400	Critical/warning
Overall Detection Rate	92.5%	370/400	Detected
Response Time	< 1 second	Real-time	Real-time

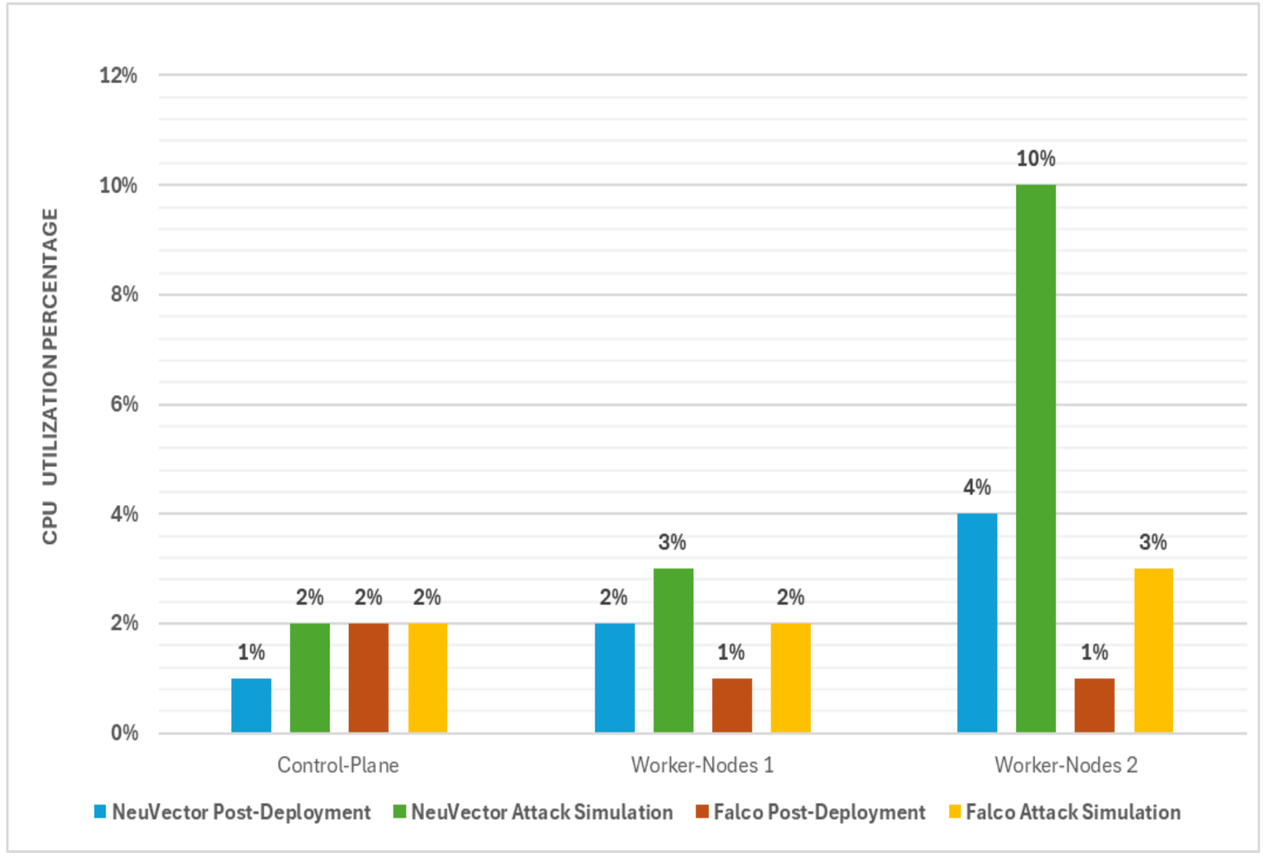
False Positive Interpretation

- Falco false positives were largely driven by legitimate Flannel CNI communication.
- Falco was evaluated with the default rule set used in the experiment; rule tuning was not the focus of this study.
- NeuVector and Falco use different monitoring models, so alert behavior differs.
- The reported FPR reflects the evaluated experimental setup and workload, not all real-world deployments.

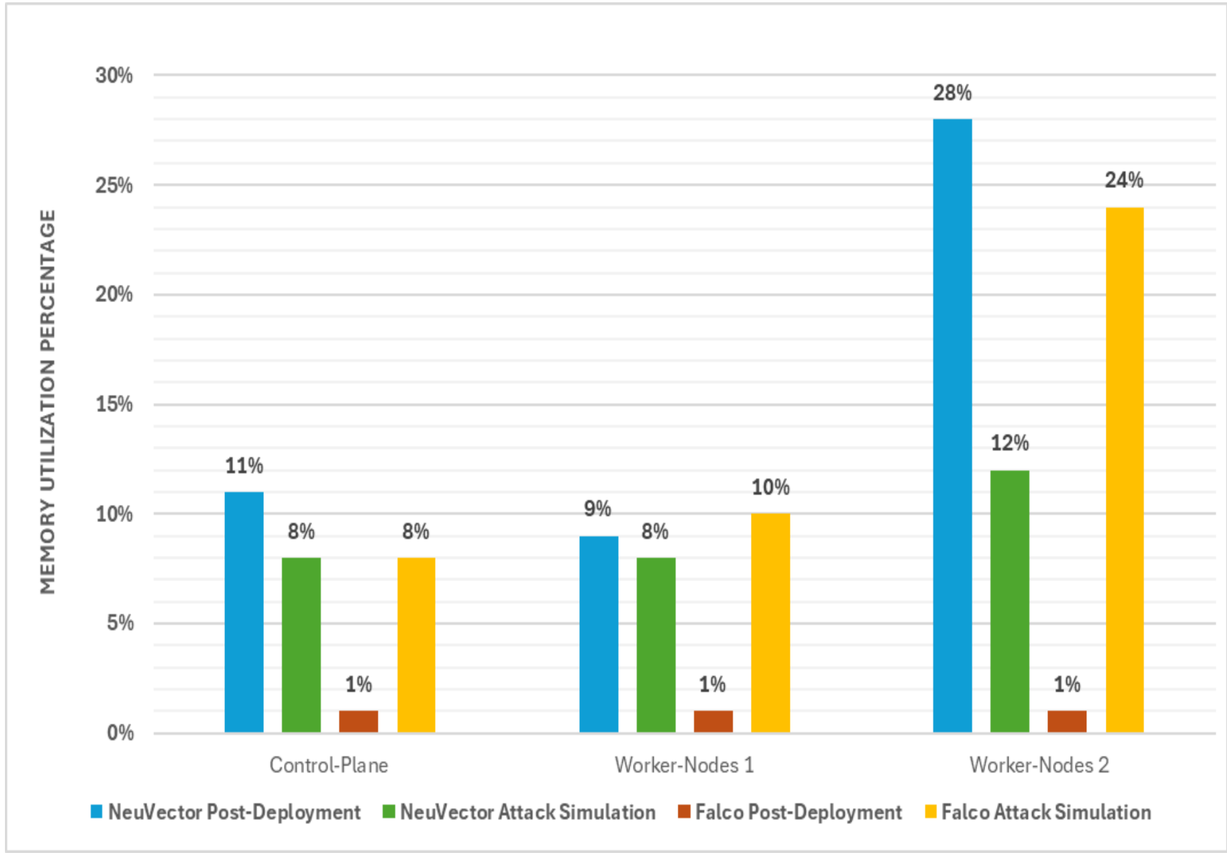
Comparative Analysis

NeuVector	Falco
Higher overhead	Lower overhead
Detection rate: 96.7%	Detection rate: 92.5%
FPR: 10.7%	FPR: 82.1%
Broader security capabilities	Lightweight runtime monitoring

Comparative Analysis

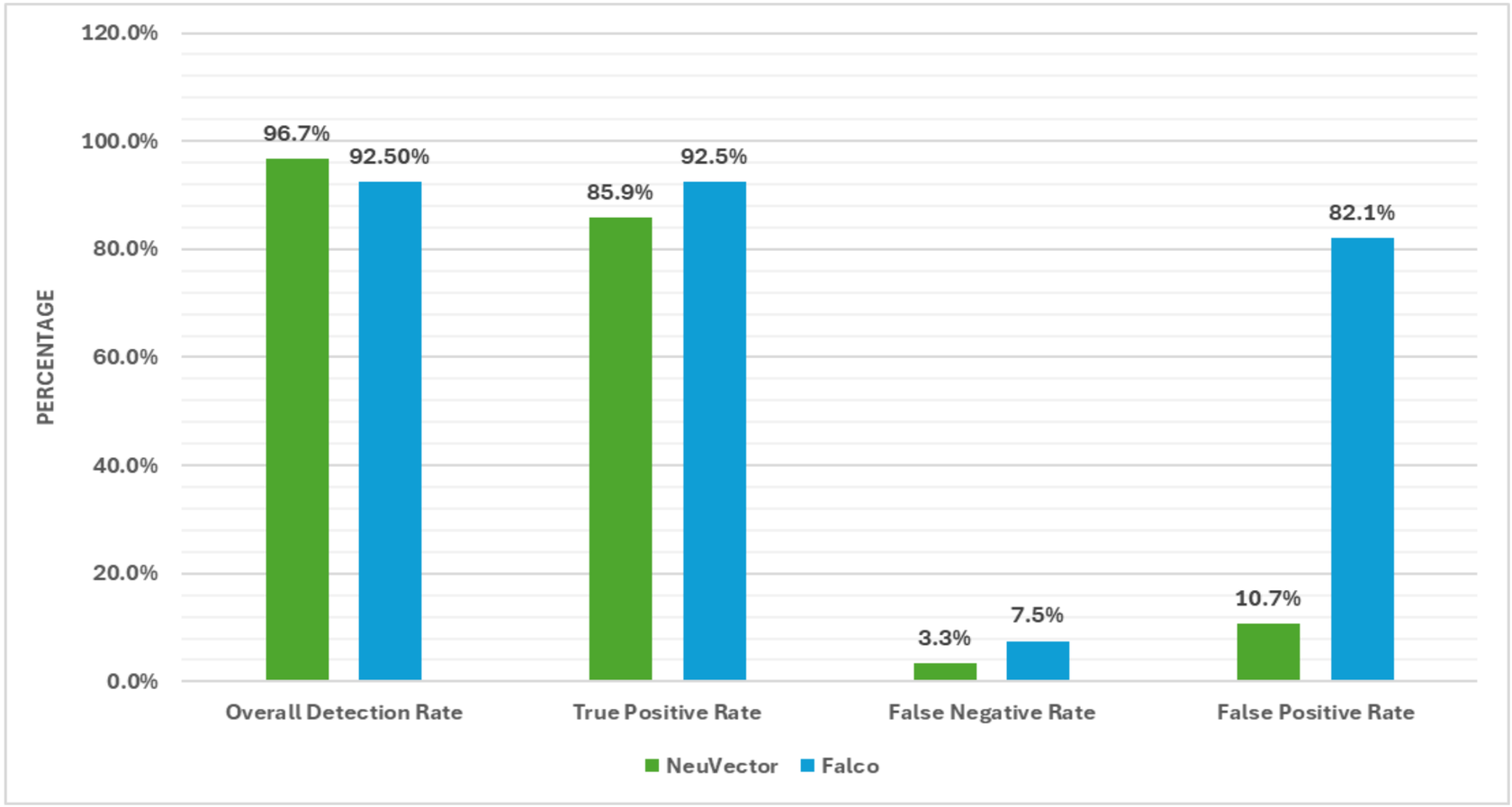


CPU Overhead (Δ) Analysis Under Deployment and Attack Conditions



Memory Overhead (Δ) Analysis Under Deployment and Attack Conditions

Comparative Analysis



Security Effectiveness Comparison NeuVector Vs. Falco

Conclusion

Key Findings

- Both tools maintained stable 5G service operation during evaluation
- NeuVector achieved 96.7% overall detection with lower false positives
- Falco achieved 92.5% detection with lower resource overhead
- NeuVector consumed substantially more CPU and memory than Falco
- Falco requires rule tuning due to high false positives
- Tool selection depends on deployment priorities: protection depth Vs. resource efficiency

Future Work

- Study multi-stage attack simulations
- Investigate more realistic 5G-specific attack scenarios
- Explore machine-learning-based false-positive reduction
- Evaluate under higher baseline utilization and production-like conditions
- Analyze SBA exploitation and network slicing isolation bypass attempts
- Improve detection accuracy and operational efficiency, particularly for Falco

Thank You!

Musab M. W. MohamedAli

Università di Genova