Cloudflare Uses eBPF to Power Security, Networking, and Observability at Global Scale

OVERVIEW

Cloudflare operates one of the world's largest edge networks, delivering security, performance, and reliability services across hundreds of cities globally. The company handles massive volumes of Internet traffic while defending against some of the most sophisticated DDoS attacks ever recorded, making traditional Linux networking and observability approaches insufficient.
To address these limitations, Cloudflare adopted eBPF as a foundational platform technology across its networking, security, and observability stack. Today, Cloudflare uses eBPF for high-performance DDoS mitigation, programmable socket dispatch, deep packet inspection, distributed tracing, load balancing, and zero-downtime service upgrades. By extending the Linux kernel safely and dynamically, eBPF has enabled Cloudflare to improve performance, reduce infrastructure complexity, and deploy new capabilities across its global edge network.

CHALLENGE

Cloudflare's edge architecture operates at enormous scale, processing massive amounts of Internet traffic while defending against increasingly sophisticated attacks. This created several technical and operational challenges:

- **Traditional Linux networking APIs** and socket models were not designed for Cloudflare's Anycast architecture or to handle the scale needed to run millions of IP addresses, without workarounds.
- **Existing DDoS** mitigation approaches relied on hardware-dependent or kernel-bypass technologies that limited hardware flexibility and operational consistency.
- **Standard observability** tools lacked visibility into kernel-level behavior and could not expose the latency distributions or packet flows needed to diagnose complex incidents.

Cloudflare needed the ability to update networking and security logic dynamically across its global edge network without downtime or kernel instability.
These limitations pushed Cloudflare to seek a programmable, high-performance, and vendor-neutral approach that could unify networking, security, and observability functions within the Linux kernel.

SOLUTION

Cloudflare adopted eBPF as a platform-level technology integrated throughout its edge infrastructure. Key implementations include:

• **L4Drop (XDP-based DDoS mitigation)**: Drops malicious traffic at the earliest point in the network stack using XDP.
• Tubular (Programmable socket dispatch): Enables dynamic socket routing via the sk_lookup eBPF hook.
• **Magic Firewall:** Embeds eBPF programs for advanced packet inspection and policy enforcement.
• **ebpf_exporter:** Collects high-resolution kernel metrics, latency histograms, and tracing data.
• **udpgrm and flowtrackd**: Powers zero-downtime UDP restarts and stateful TCP flow tracking.

RESULTS

Cloudflare's use of eBPF has produced significant operational, performance, and architectural benefits:

- **Record-Scale Autonomous Mitigation**: The system mitigated a record-breaking 31.4 Tbps DDoS attack in November 2025 and a packet-rate attack of 14.1 billion packets per second in October 2025, both without human intervention.
- **High-performance DDoS mitigation**: Cloudflare's XDP-based L4Drop system can drop more than 10 million packets per second on a single CPU core, while maintaining low CPU overhead during large attacks.
- **Faster autonomous incident response**: Local detection via dosd deploys mitigation rules in seconds rather than minutes. Engineers can now diagnose kernel race conditions and "invisible" latency spikes of 500ms+.
- **Reduced infrastructure complexity**: eBPF allowed Cloudflare to replace multiple legacy systems, custom kernel patches, and hardware-dependent solutions with a unified, software-defined platform.
- **Improved hardware flexibility**: Moving from proprietary kernel-bypass technologies to eBPF-enabled XDP reduced hardware lock-in and improved compatibility across multiple NIC vendors and server architectures.
- **Advanced observability**: eBPF-based tracing and histogram collection enabled Cloudflare engineers to diagnose issues that traditional tools could not detect, including kernel race conditions, packet corruption, hidden latency spikes, and fragmentation buffer exhaustion.
- **Scalable IP management**: Cloudflare used eBPF-based load balancing to implement "soft-unicast," allowing a single IPv4 address to be shared across dozens of physical servers and helping mitigate IPv4 address exhaustion.

Today, eBPF powers critical networking, observability, and security services across Cloudflare's global edge infrastructure.

FUTURE PLANS

Cloudflare continues to expand its use of eBPF across multiple domains. Areas of active development include:
• Rich packet metadata infrastructure to improve end-to-end packet tracing and service identification.
• Expanded deep packet inspection and protocol validation within Magic Firewall.
• Performance improvements for BPF data structures such as the LPM trie.
• Programmable Flow Protection capabilities for custom eBPF traffic filtering.
• Broader adoption of eBPF-powered graceful restart technologies for UDP-based protocols.
Cloudflare also plans to continue contributing improvements upstream to the Linux kernel and the broader eBPF ecosystem.

CONCLUSION

WHY eBPF?

Cloudflare selected eBPF because it provides a safe, programmable, and high-performance way to extend Linux kernel functionality without custom kernel modules.

eBPF enables Cloudflare to:
• Execute networking and security logic directly in the kernel with low overhead.
• Dynamically update functionality without rebooting servers.
• Safely deploy programmable packet processing through the kernel verifier.
• Build modular and composable networking pipelines.
• Create unified tooling across security, observability, and networking domains.