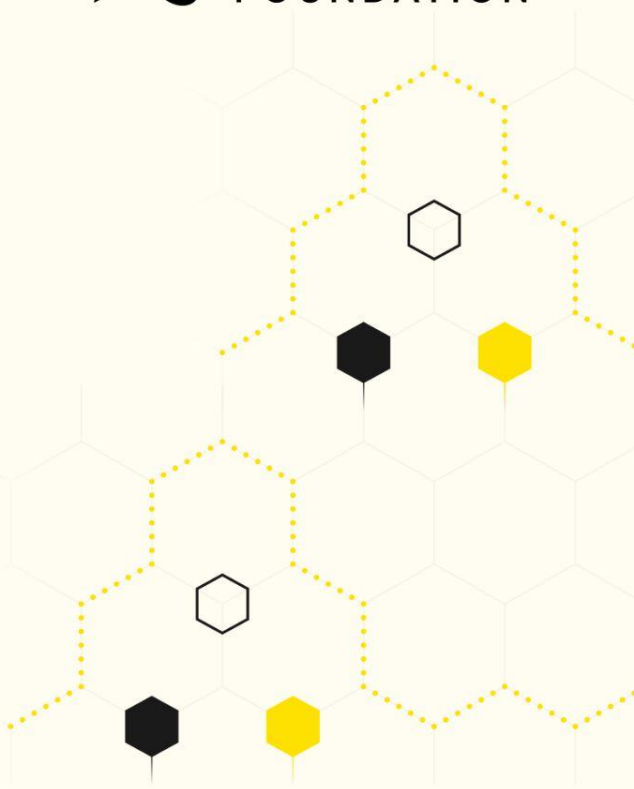
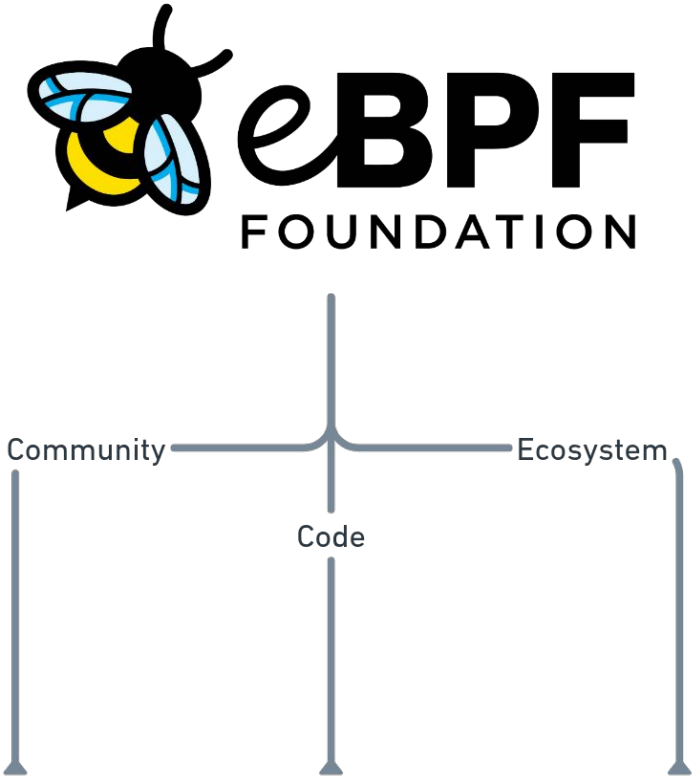
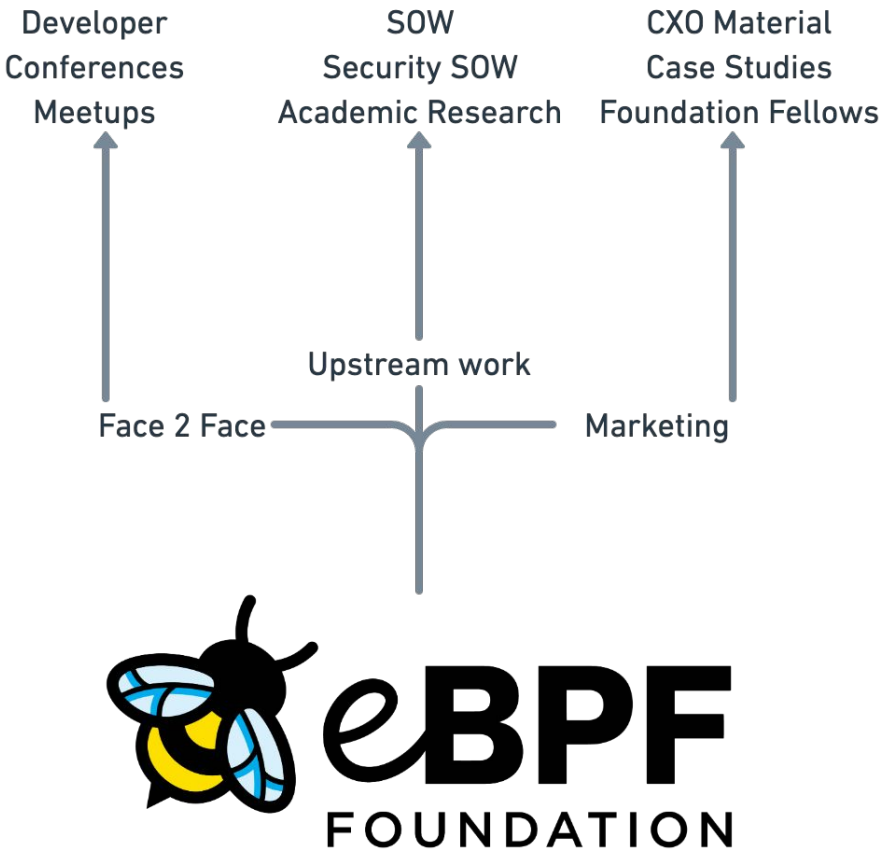
From Projects to Ecosystems: Lessons from the eBPF Foundation



To innovate and enable others to innovate



eBPF Foundation Strategy



Developer Face to Face

Linux Kernel Developers' bpfconf 2025

bpfconf is an invitation-only technical workshop run by the Linux community in order to bring BPF core developers together, to discuss new ideas and to work out improvements to the BPF subsystem that will make their way into future mainline kernels and into the BPF compiler backends.

The conference is purposely kept small with focus on discussion rather than just presentation. Along with the LPC's BPF Track which is organized and run by the same community, the goal is to allow developers to meet face to face twice per year to exchange and discuss ongoing developments in the BPF ecosystem.

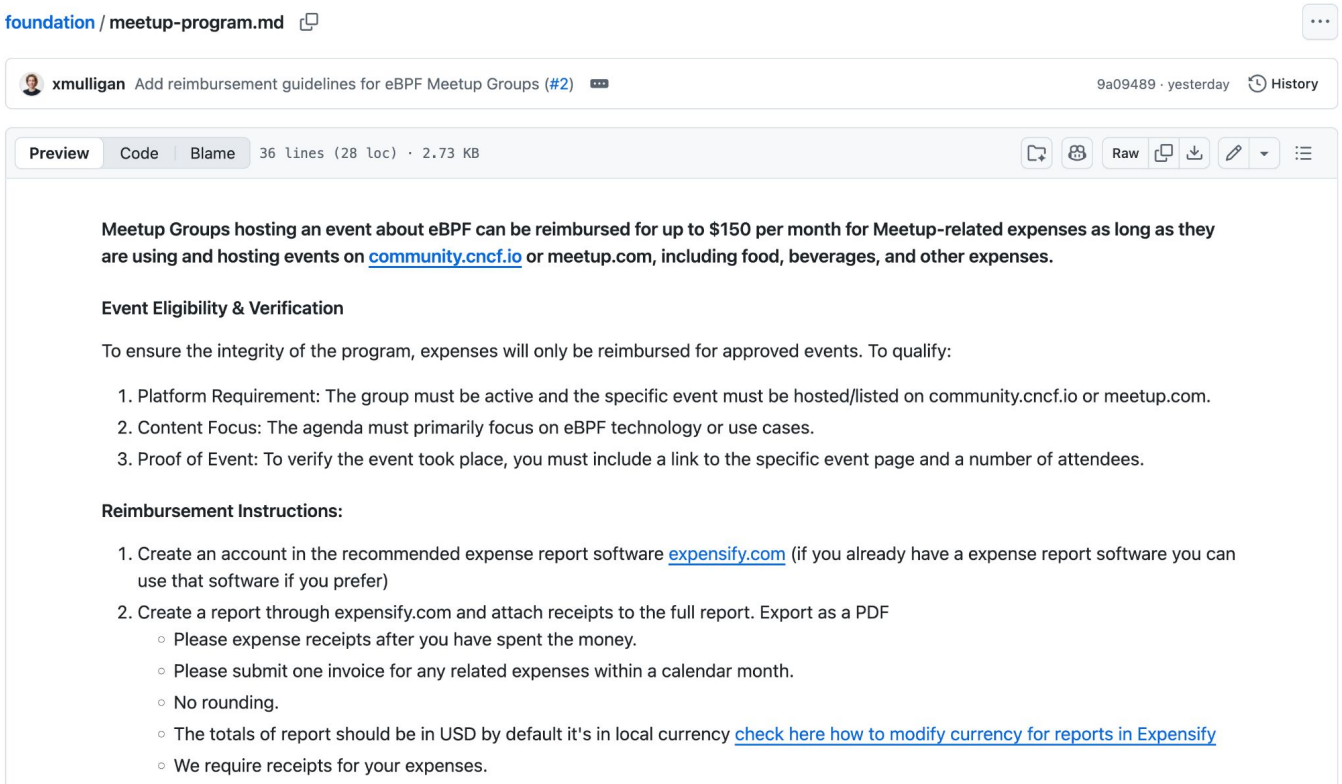
The 2025 bpfconf edition is a three-days conference which is part of the LSF/MM/BPF summit. It is therefore also open to all LSF/MM/BPF attendees.

Discussion Topics

The following discussion topics have been brought up at this year's bpfconf. In each slot below, there is a short discussion topic with a link to the corresponding slides in case slides have been used as a discussion starter.

LWN coverage of the BPF track from Jonathan Corbet can be found here.

Developer Face to Face



foundation / meetup-program.md

xmulligan Add reimbursement guidelines for eBPF Meetup Groups (#2) 9a09489 · yesterday History

Preview Code Blame 36 lines (28 loc) · 2.73 KB Raw

Meetup Groups hosting an event about eBPF can be reimbursed for up to $150 per month for Meetup-related expenses as long as they are using and hosting events on community.cncf.io or meetup.com, including food, beverages, and other expenses.

Event Eligibility & Verification

To ensure the integrity of the program, expenses will only be reimbursed for approved events. To qualify:

1. Platform Requirement: The group must be active and the specific event must be hosted/listed on community.cncf.io or meetup.com.
2. Content Focus: The agenda must primarily focus on eBPF technology or use cases.
3. Proof of Event: To verify the event took place, you must include a link to the specific event page and a number of attendees.

Reimbursement Instructions:

1. Create an account in the recommended expense report software expensify.com (if you already have a expense report software you can use that software if you prefer)
2. Create a report through expensify.com and attach receipts to the full report. Export as a PDF
   - Please expense receipts after you have spent the money.
   - Please submit one invoice for any related expenses within a calendar month.
   - No rounding.
   - The totals of report should be in USD by default it's in local currency check here how to modify currency for reports in Expensify
   - We require receipts for your expenses.

* Meetup program coming soon

Upstream Development SOW

index : kernel/git/torvalds/linux.git

Linux kernel source tree

master switch

Linus Torvalds

about summary refs log tree commit diff stats

log msg eBPF Fou search

Age	Commit message (Expand)	Author	Files	Lines
6 days	Merge branch 'selftests-bpf-convert-test_tc_edt-sh-into-test_progs'	Alexei Starovoitov	4	-107/+151
6 days	selftests/bpf: do not hardcode target rate in test_tc_edt BPF program	Alexis Lothoré (eBPF Foundation)	2	-3/+4
6 days	selftests/bpf: remove test_tc_edt.sh	Alexis Lothoré (eBPF Foundation)	2	-102/+0
6 days	selftests/bpf: integrate test_tc_edt into test_progs	Alexis Lothoré (eBPF Foundation)	2	-1/+145
6 days	selftests/bpf: rename test_tc_edt.bpf.c section to expose program type	Alexis Lothoré (eBPF Foundation)	2	-2/+3
2025-11-06	selftests/bpf: Use start_server_str rather than start_reuseport_server in tc_...	Alexis Lothoré (eBPF Foundation)	1	-15/+13
2025-11-06	selftests/bpf: Systematically add SO_REUSEADDR in start_server_addr	Alexis Lothoré (eBPF Foundation)	1	-1/+7
2025-10-31	selftests/bpf: Add checks in tc_tunnel when entering net namespaces	Alexis Lothoré (eBPF Foundation)	1	-46/+88
2025-10-31	selftests/bpf: Skip tc_tunnel subtest if its setup fails	Alexis Lothoré (eBPF Foundation)	1	-2/+2
2025-10-31	Merge branch 'selftests-bpf-integrate-test_xsk-c-to-test_progs-framework'	Alexei Starovoitov	6	-2632/+3092
2025-10-31	selftests/bpf: test_xsk: Integrate test_xsk.c to test_progs framework	Bastien Curutchet (eBPF Foundation)	5	-3/+163
2025-10-31	selftests/bpf: test_xsk: Isolate non-CI tests	Bastien Curutchet (eBPF Foundation)	2	-17/+32
2025-10-31	selftests/bpf: test_xsk: Don't exit immediately on allocation failures	Bastien Curutchet (eBPF Foundation)	3	-42/+110
2025-10-31	selftests/bpf: test_xsk: Don't exit immediately if validate_traffic fails	Bastien Curutchet (eBPF Foundation)	1	-4/+8
2025-10-31	selftests/bpf: test_xsk: Don't exit immediately when workers fail	Bastien Curutchet (eBPF Foundation)	1	-34/+76
2025-10-31	selftests/bpf: test_xsk: Don't exit immediately when gettimeofday fails	Bastien Curutchet (eBPF Foundation)	1	-4/+4
2025-10-31	selftests/bpf: test_xsk: Don't exit immediately when xsk_attach fails	Bastien Curutchet (eBPF Foundation)	1	-9/+19
2025-10-31	selftests/bpf: test_xsk: Add return value to init_iface()	Bastien Curutchet (eBPF Foundation)	3	-6/+11
2025-10-31	selftests/bpf: test_xsk: Release resources when swap fails	Bastien Curutchet (eBPF Foundation)	1	-1/+6
2025-10-31	selftests/bpf: test_xsk: Wrap test clean-up in functions	Bastien Curutchet (eBPF Foundation)	1	-12/+24
2025-10-31	selftests/bpf: test_xsk: fix memory leak in testapp_xdp_shared_umem()	Bastien Curutchet (eBPF Foundation)	1	-1/+22
2025-10-31	selftests/bpf: test_xsk: fix memory leak in testapp_stats_rx_dropped()	Bastien Curutchet (eBPF Foundation)	1	-0/+7
2025-10-31	selftests/bpf: test_xsk: Fix __testapp_validate_traffic()'s return value	Bastien Curutchet (eBPF Foundation)	1	-1/+4
2025-10-31	selftests/bpf: test_xsk: Initialize bitmap before use	Bastien Curutchet (eBPF Foundation)	1	-0/+4
2025-10-31	selftests/bpf: test_xsk: Split xskxceiver	Bastien Curutchet (eBPF Foundation)	5	-2634/+2738
2025-10-29	selftests/bpf: Remove test_tc_tunnel.sh	Alexis Lothoré (eBPF Foundation)	2	-321/+0
2025-10-29	selftests/bpf: Integrate test_tc_tunnel.sh tests into test_progs	Alexis Lothoré (eBPF Foundation)	2	-19/+693
2025-10-29	selftests/bpf: Make test_tc_tunnel.bpf.c compatible with big endian platforms	Alexis Lothoré (eBPF Foundation)	1	-35/+22
2025-10-29	selftests/bpf: Add tc helpers	Alexis Lothoré (eBPF Foundation)	3	-93/+74

Upstream Development

index : kernel/git/torvalds/linux.git

Linux kernel source tree

about summary refs log tree commit diff stats

author	Alexis Lothoré (eBPF Foundation) <alexis.lothore@bootlin.com>	2025-11-28 23:27:19 +0100
committer	Alexei Starovoitov <ast@kernel.org>	2025-11-29 09:37:41 -0800
commit	b0f82e7ab6fb2f8501ef87ae928cbf7358d7845e (patch)	
tree	b349f81a8b8e95c9b5241b8e771c14e774222e36	
parent	4b4833acc63e9c8ea9d5897ee84b694f30b23882 (diff)	
download	linux-b0f82e7ab6fb2f8501ef87ae928cbf7358d7845e.tar.gz	

selftests/bpf: integrate test_tc_edt into test_progs

```
test_tc_edt.sh uses a pair of veth and a BPF program attached to the TX
veth to shape the traffic to 5MBps. It then checks that the amount of
received bytes (at interface level), compared to the TX duration, indeed
matches 5Mbps.

Convert this test script to the test_progs framework:
- keep the double veth setup, isolated in two veths
- run a small tcp server, and connect client to server
- push a pre-configured amount of bytes, and measure how much time has
  been needed to push those
- ensure that this rate is in a 2% error margin around the target rate

This two percent value, while being tight, is hopefully large enough to
not make the test too flaky in CI, while also turning it into a small
example of BPF-based shaping.

Signed-off-by: Alexis Lothoré (eBPF Foundation) <alexis.lothore@bootlin.com>
Link: https://lore.kernel.org/r/20251128-tc_edt-v2-2-26db48373e73@bootlin.com
Signed-off-by: Alexei Starovoitov <ast@kernel.org>
```

Diffstat

-rw-r--r--	tools/testing/selftests/bpf/prog_tests/test_tc_edt.c	144
-rw-r--r--	tools/testing/selftests/bpf/progs/test_tc_edt.c	2

2 files changed, 145 insertions, 1 deletions

```
diff --git a/tools/testing/selftests/bpf/prog_tests/test_tc_edt.c b/tools/testing/selftests/bpf/prog_tests/test_tc_edt.c
new file mode 100644
index 00000000000000..9ba69398eec49c
--- /dev/null
+++ b/tools/testing/selftests/bpf/prog_tests/test_tc_edt.c
@@ -0,0 +1,144 @@
```

bootlin

Home Engineering Training Docs

Improving the eBPF tests in the kernel

Alexis Lothoré

March 17, 2025

Technical

ebpf, testing, upstream

As part of a partnership with the eBPF Foundation, Bootlin engineers Bastien Curutchet and Alexis Lothoré are working with the kernel community in order to improve eBPF support in the kernel on different aspects. This post is the first one of a series highlighting this effort. For those who need to catch up with the eBPF technology, you can take a look at our "Linux Debugging, tracing and profiling" training course which has been recently updated with eBPF basics !

Upstream Development - BTF, Trampolines, and CI

- **Runtime Type Information via BTF**: Solved DWARF size constraints (reducing ~200MB to ~5MB) allowing embedding runtime-loadable type data for dynamic function modeling
- **Low-Overhead Trampolines**: Replaced expensive kprobes with direct function calls acting as a dynamic ABI bridge for fentry/fexit
- **ARM64 Feature Parity**: Multi-kprobes and complex argument handling for functions with >8 parameters (stack-passed args)
- **Test Suite Modernization**: Migrated legacy standalone shell scripts to the generic test_progs C runner
- **Expanded CI Coverage**: Increased automated regression testing reliability on bpf-next

What SOWs should we fund in the future?

- riscv64 BPF JIT on par with x86-64 & arm64
- XXX

Upstream Security Funding

nccgroup

eBPF Verifier Code Review

NCC Group
Version 1.0 – November 11, 2024

eBPF FOUNDATION

Prepared By
Chris Anley
Nathaniel Theis

Prepared For
eBPF Foundation

5 Finding Details

High

find_equal_scalars Mishandles 32-Bit Addition

Overall Risk	High	**Finding ID**	NCC-E015561-JJX
Impact	High	**Component**	eBPF verifier
Exploitability	Low	**Category**	Data Validation
		Status	Fixed

Impact
An attacker with CAP_BPF (required to reach the vulnerable code paths) who can load & execute eBPF programs on a vulnerable system can read to, and write from, arbitrary kernel memory.

Description
A vulnerability in the eBPF verifier permits an attacker to submit and run an eBPF program that can read from, and write to, arbitrary kernel memory.

The issue involves the tracking of eBPF register values, and specifically the identification of equal scalars. The specific vulnerable code is related to the `find_equal_scalars()` function and the BPF_ADD_CONST flag, which signifies a constant offset between two scalar registers.

An attacker requires either root privilege or CAP_BPF[2] to successfully exploit the issue. Additionally, the POC code below requires CAP_PERFMON, because it doesn't bypass the ALU sanitizer (although in an actual exploit this can be achieved with previously documented[3] methods).

The verifier attempts to track "similar" scalars in order to propagate bounds information learned about one scalar to others. For instance, if `r1` and `r2` are known to contain the same value, then upon encountering `if (r1 != 0x1234) goto 1234;`, not only does it know that `r1` is equal to 0x1234 on the path where that conditional jump is not taken, it also knows that `r2` is.

Additionally, if `env->bpf_capable` (i.e. if the process loading this eBPF program has CAP_BPF), the verifier will track scalars which should be a constant delta apart (if `r1` is known to be one greater than `r2`, then if `r1` is known to be equal to 0x1234, `r2` must be equal to 0x1233.)

The relevant code from `adjust_reg_min_max_vals()`, located at kernel/bpf/verifier.c:14101:

```
if (env->bpf_capable && BPF_OP(insn->code) == BPF_ADD &&
      dst_reg->id && is_reg_const(src_reg, alu32)) {
    u64 val = reg_const_value(src_reg, alu32);

    if ((dst_reg->id & BPF_ADD_CONST) ||
        /* prevent overflow in find_equal_scalars() later */
        val > (u32)S32_MAX) {
```

2. CAP_BPF is a Linux kernel capability which serves several purposes. First, it allows certain eBPF operations such as creating maps. Second, it enables more sophisticated analysis in the eBPF verifier (including the vulnerable scalar-offset analysis documented in this finding). Third, if unprivileged eBPF is disabled by the `unprivileged_bpf_disabled` sysctl- as it currently is on most popular distributions by default- CAP_BPF is required to load any BPF program.
3. A technique is described in this writeup by Manfred Paul of his winning entry in the Pwn2Own 2020 competition.

eBPF

Security Threat Model

Jack Kelly, James Callaghan & Andrew Martin

control-plane.io

Upstream Security Funding

$225,000 grant from Alpha-Omega for:

- Defensive runtime tooling
  - **Enabling KASAN** for JIT-compiled eBPF programs to validate memory accesses
- Auditing
  - **Security review of x86-64, arm64, and riscv64 JIT compilers**, specifically targeting vulnerabilities in instruction encoding, register allocation, and immediate value handling
  - **Check verifier-to-JIT integrity** ensuring security assumptions (memory boundaries, control flow) are preserved in translation
  - **Assessment of unprivileged surfaces**, cBPF to eBPF translation (bpf_convert_filter) and seccomp filter validation

How can we improve eBPF security?

- XXX

Academic Research Grants

Announcements Blog

eBPF Foundation Announces $250,000 in Grant Awards for Five eBPF Academic Research Projects

By Dan Brown | August 29, 2024 | 7 min read

Projects will advance eBPF's open source technology by improving scalability, static an and more

Announcements Blog

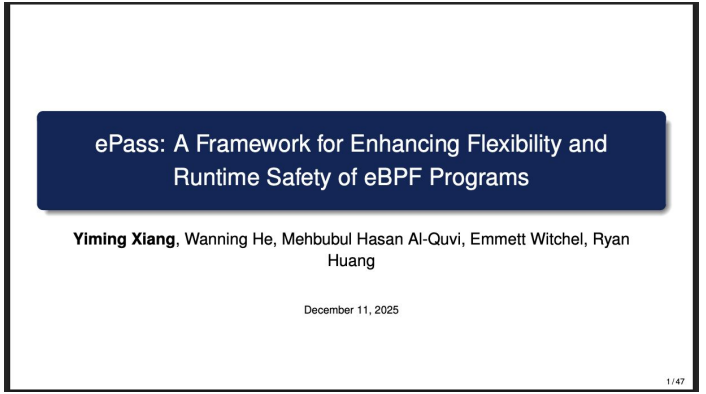
eBPF Foundation Awards $100,000 in Research Grants to Advance eBPF Safety and Efficiency

By Dan Brown | September 16, 2025 | 4 min read

Research projects will strengthen eBPF programmability, runtime safety, and datacenter energy efficiency through verifier-cooperative instrumentation and QoS-aware power management

Academic Research Grants

- Enhancing Flexibility and Safety of eBPF Programs through Verifier-Cooperative Instrumentation
- eBPF Governors: QoS-aware Agile Power Management



- Efficient IO-Intensive µs-scale Applications using eBPF
- Verified Path Exploration for eBPF Verifier Static Analysis
- Lazy Abstraction Refinement with Proof for an Enhanced Verifier
- Improving eBPF Complexity with a Hardware-backed Isolation Environment
- Learned Virtual Memory with eBPF

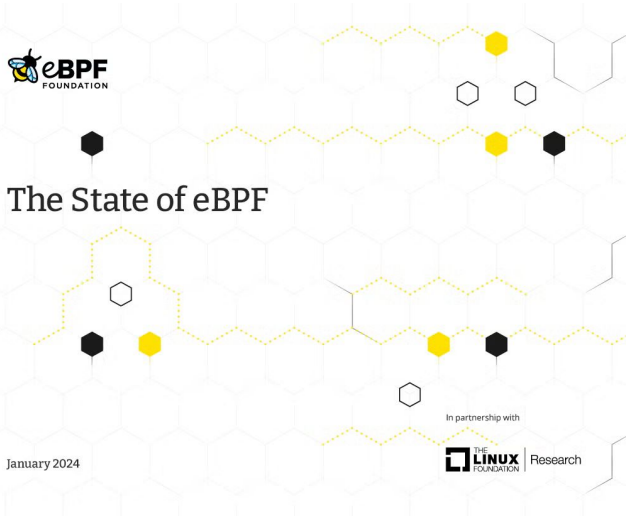
What research is important?

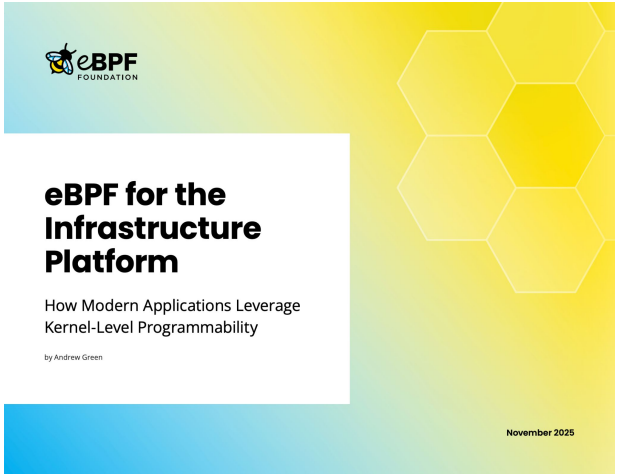
- XXX

CXO Education Material

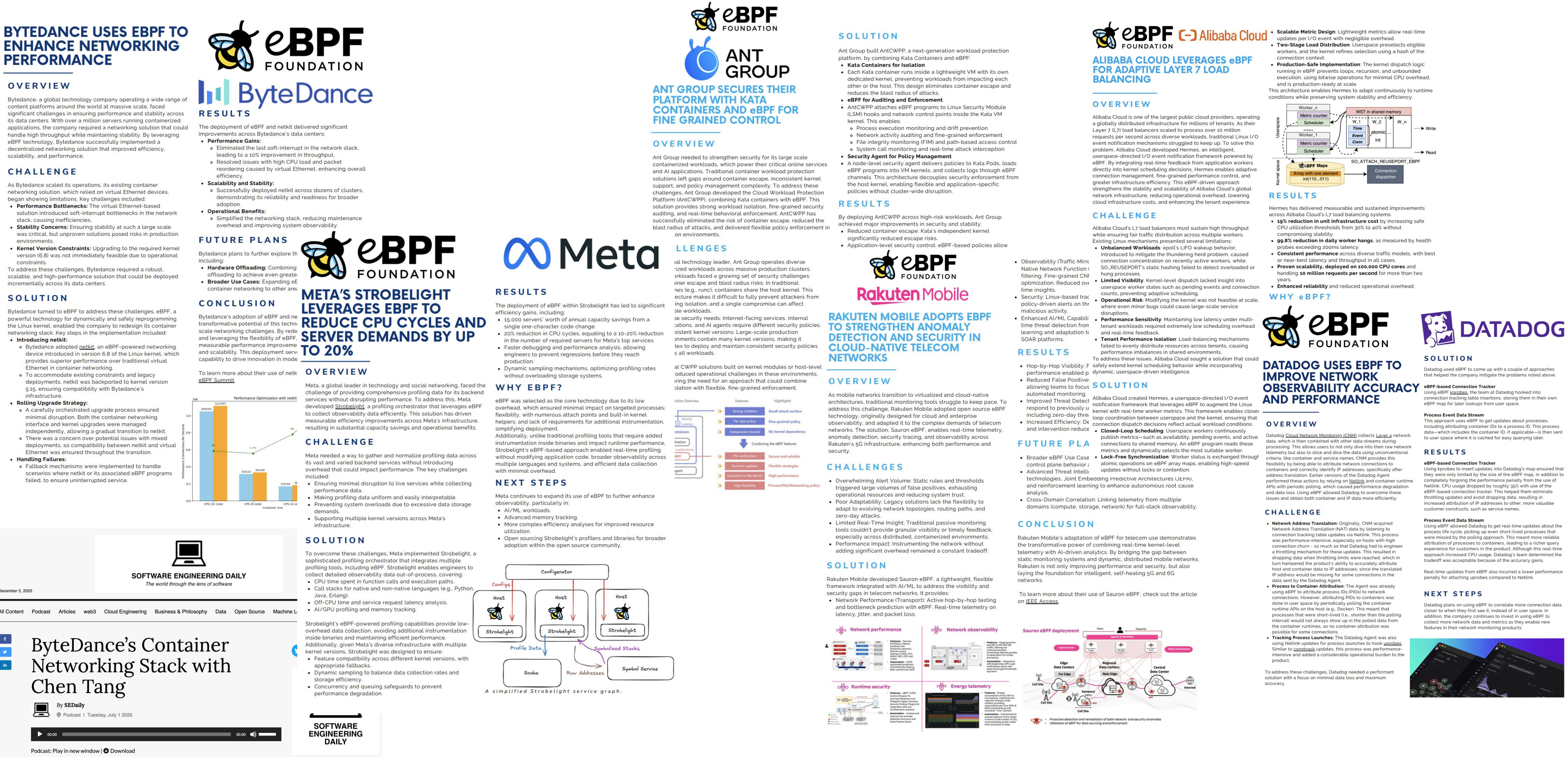
eBPF: Unlocking the Kernel [OFFICIAL DOCUMENTARY]

150K views • 1 year ago





Case Studies - Your company next?



Community Building Fellowships

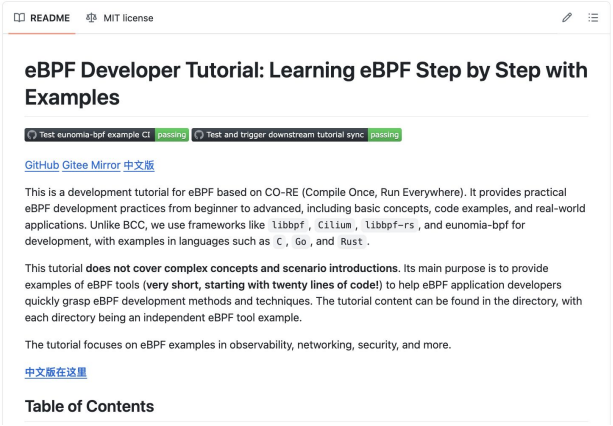
Uncategorized

eBPF Foundation Announces Inaugural Community & Advocacy Fellowship Recipients

By Dan Brown | October 15, 2025 | 4 min read

TL;DR

The **eBPF Foundation** announced the recipients of its inaugural **Community & Advocacy Fellowship**, recognizing contributors expanding global access to eBPF through education, tutorials, and community-building. The 2025 fellows are **Teodor Janez Podobnik** (eBPFChirp Newsletter) and **Yusheng Zheng** (bpf-developer-tutorial).


README MIT license

eBPF Developer Tutorial: Learning eBPF Step by Step with Examples

Test eunomia-bpf example CI passing Test and trigger downstream tutorial sync passing

GitHub Gitee Mirror 中文版

This is a development tutorial for eBPF based on CO-RE (Compile Once, Run Everywhere). It provides practical eBPF development practices from beginner to advanced, including basic concepts, code examples, and real-world applications. Unlike BCC, we use frameworks like libbpf, Cilium, libbpf-rs, and eunomia-bpf for development, with examples in languages such as C, Go, and Rust.

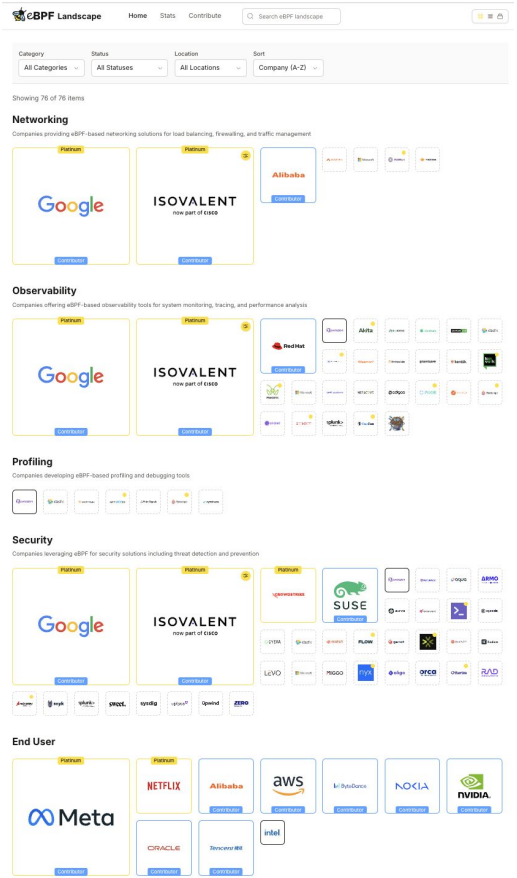
This tutorial **does not cover complex concepts and scenario introductions**. Its main purpose is to provide examples of eBPF tools (**very short, starting with twenty lines of code!**) to help eBPF application developers quickly grasp eBPF development methods and techniques. The tutorial content can be found in the directory, with each directory being an independent eBPF tool example.

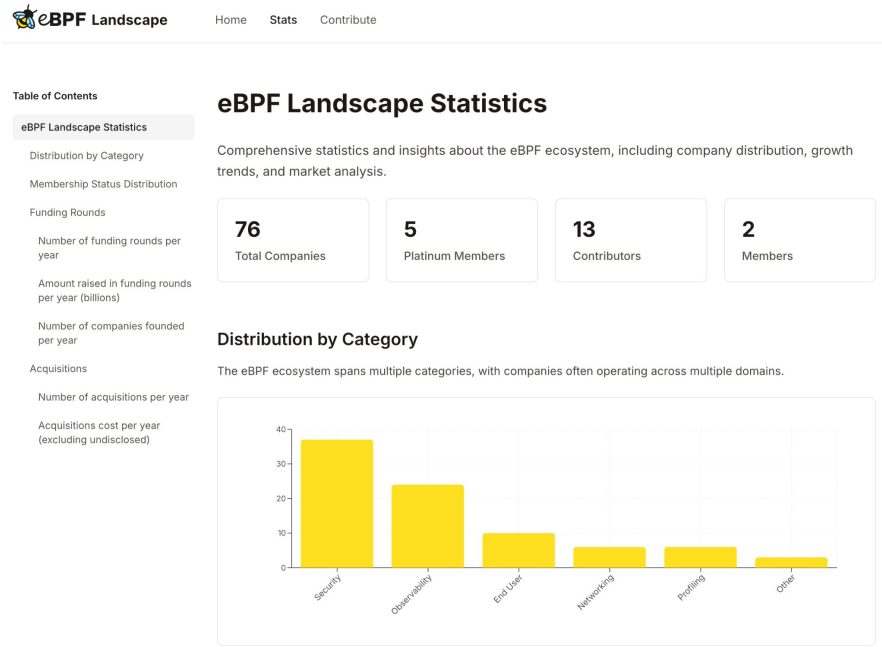
The tutorial focuses on eBPF examples in observability, networking, security, and more.

中文版在这里

Table of Contents

Company Landscape





What information would be helpful to surface?

- XXX

Join the Foundation

eBPF FOUNDATION

Who are our members

Platinum

Silver

Join the Foundation

How you benefit from membership

Strategic Influence

Take part in shaping the technical roadmap and key decisions for the future of eBPF.

Cross-Platform Collaboration

Work openly with leading industry leaders and developers in a neutral, collaborative environment.

Security & Research

Contribute to security audits, threat modeling, and academic research to strengthen eBPF.

Visibility & Credibility

Build your organization's reputation as a trusted and visible leader in the open source ecosystem.

Flexible membership tiers for every organization

Choose the level that aligns with your organization's size and goals. All tiers include participation, visibility, and collaboration opportunities.

Silver

$5,000 – $30,000

- *5,000+ Employees – $30,000 ($10,000 For LF Members)*
- *500-4,999 Employees – $25,000 ($10,000 For LF Members)*
- *100-499 Employees – $15,000 ($5,000 For LF Members)*
- *Up to 99 Employees – $10,000 ($5,000 For LF Members)*

Get started

Shared influence, training perks, marketing access, and brand visibility.

- Possible Board seat (1 per 5 Silver members)
- 10 free training tickets
- Participate in marketing & community
- Logo displayed on website & materials
- Linux Foundation Member Summit access

Platinum

$70,000

- *$50,000 if already an LF Member*

Get started

Voting rights, top visibility, leadership access, and training perks.

- 1 voting rep to Board, Marketing & subcommittees
- Premium member visibility
- Access meetings & events
- 10 free training tickets
- Quarterly membership announcement
- Direct leadership insights
- Linux Foundation Member Summit invite
- Conference & sponsorship discounts

Get in Contact

Email me: bill@ebpf.io

Email BSC: info@ebpf.foundation

eBPF Japan Community https://ebpf.connpass.com/

Discussion