Formally Verified Sandboxing for Packet Processing Programs

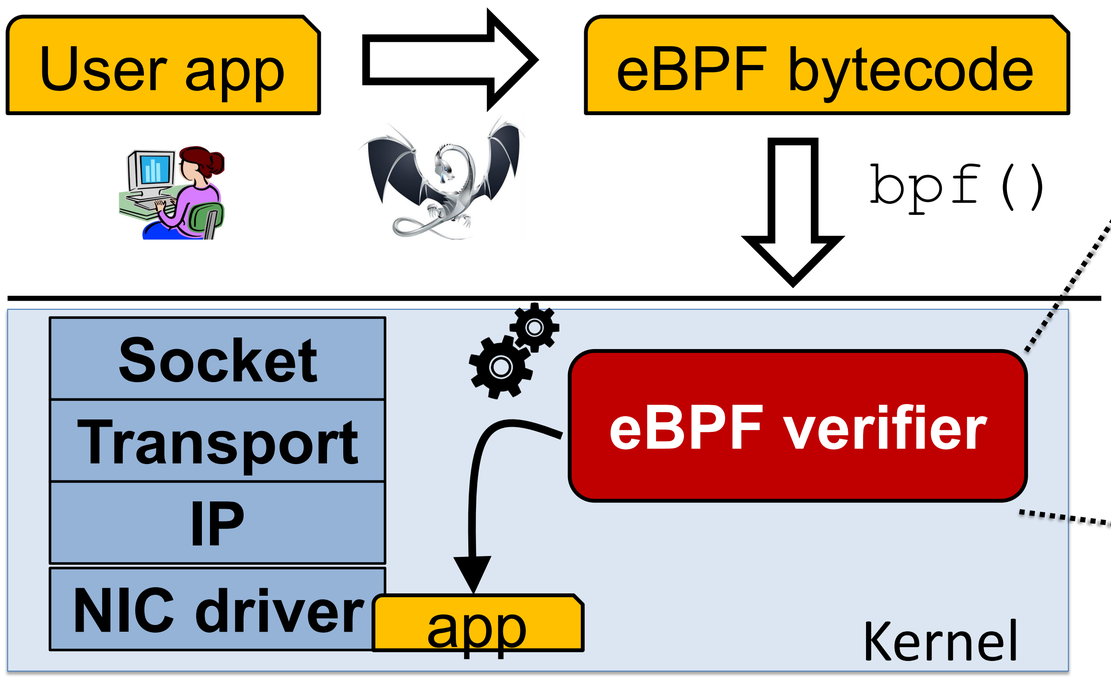
Srinivas Narayana and Santosh Nagarakatte

RUTGERS THE STATE UNIVERSITY OF NEW JERSEY

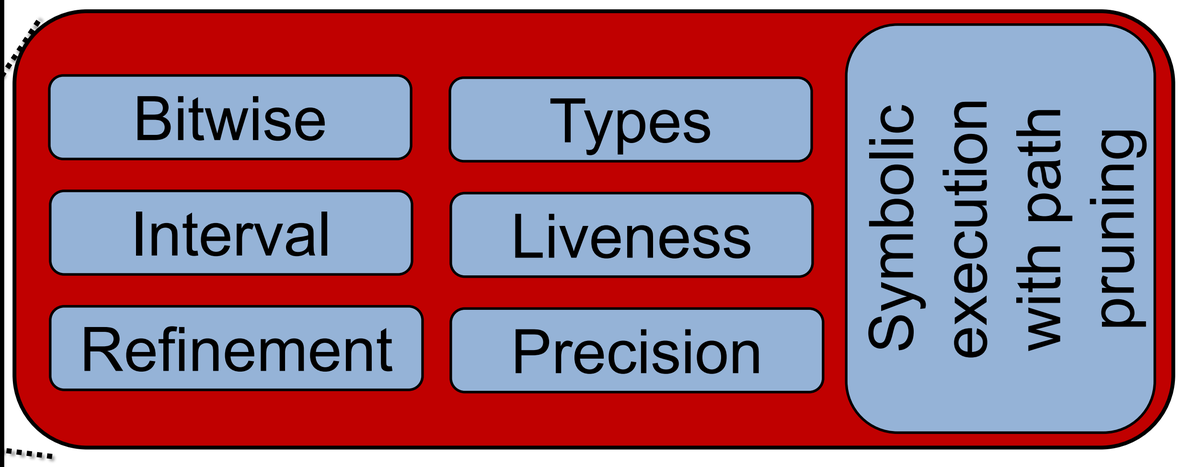
https://people.cs.rutgers.edu/~sn624/verified-sandboxing.html Award 2019302

Linux kernel extensions for networking using eBPF are widely deployed.
They are statically checked for safety before running in the kernel.
However, the in-kernel verifier has bugs, opening the kernel up to attack.
Our goal is to design a sound eBPF verifier for Linux.

Context: eBPF and Verification



Approach: Sound Abstract Interpretation



$$\forall P, Q \in \mathbb{T}_n :$$
$$\left\{ \mathbf{op}_{\mathbb{C}}(x, y) \mid x \in \gamma(P), y \in \gamma(Q) \right\} \sqsubseteq_{\mathbb{C}} \gamma(\mathbf{op}_{\mathbb{T}}(P, Q))$$

Bitwise abstractions (CGO'22)

Tri-state domain

tnum P : 10µ0 (1000, 0010)
tnum Q : 10µ1 (1001, 0010)

Prove sound+optimal add

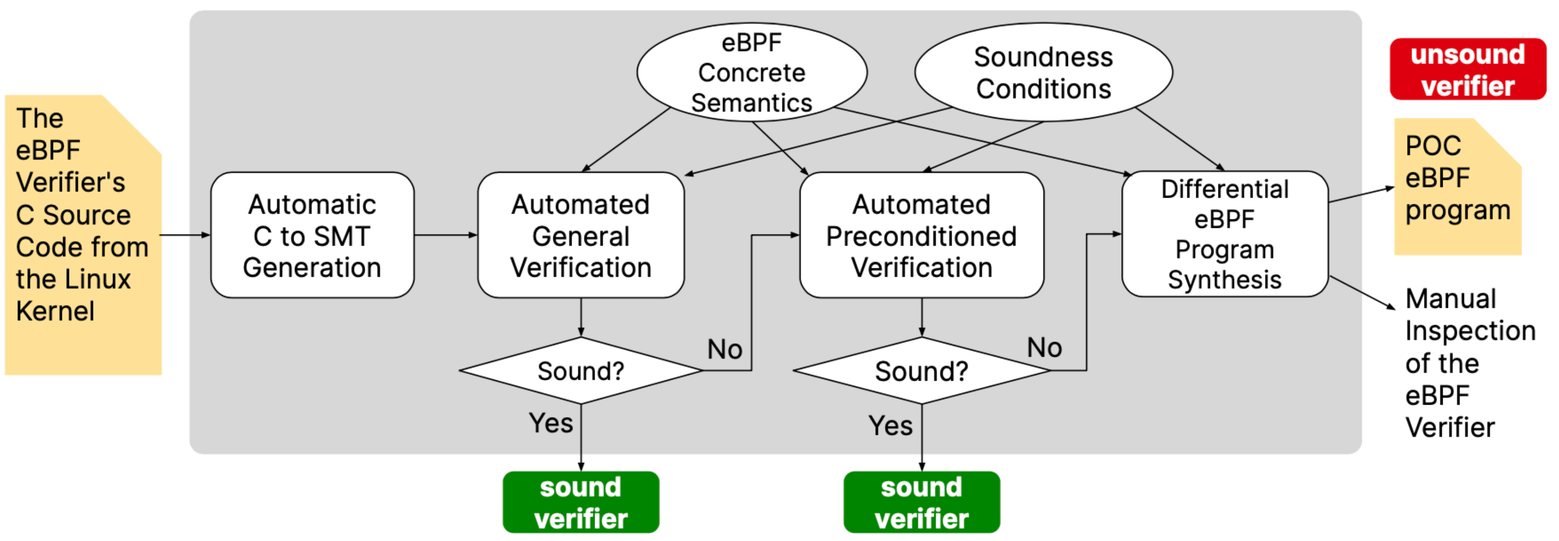
```
def tnum_add(tnum P, tnum Q):

   u64 sv := P.v + Q.v
   u64 sm := P.m + Q.m
   u64 Σ := sv + sm
   u64 χ := Σ ⊕ sv
   u64 η := χ | P.m | Q.m
   tnum R := tnum(sv & ~η, η)
   return R
```

```
def  our_mul(tnum P, tnum Q):

   ACCv := tnum(P.v * Q.v, 0)
   ACCm := tnum(0, 0)

   while P.value or P.mask:
      # LSB of tnum P is a certain 1
      if (P.v[0] == 1) and (P.m[0] == 0):
         ACCm := tnum_add(ACCm, tnum(0, Q.m))
      # LSB of tnum P is uncertain
      else if  (P.m[0] == 1):
         ACCm := tnum_add(ACCm, tnum(0, Q.v|Q.m))
      # Note: no case for LSB is certain 0
      P := tnum_rshift(P, 1)
      Q := tnum_lshift(Q, 1)

   tnum R := tnum_add(ACCv, ACCm)
   return R
```

New sound multiplication (faster, more precise than kernel's algorithm)

Intervals, multi-domain refinements (CAV'23)

Sound modular operators (SAS'24)



Formalization of multi-domain abstractions in kernel
Toolchain (Agni) to extract SMT from eBPF operators
Previously unknown bugs & simpler PoCs for existing

Broader Impacts

Technology transfer: Patches upstreamed to Linux mainline kernel (algorithm, verification fixes)
Outreach: Linux Plumbers Conference '23,'24, eBPF workshop SIGCOMM '23, '24. Actively used as CI by devs
Training: Grads (Hari Vishwanathan, Matan Shachnai) & UGs