SAND2021-8453D

The Center for Cyber Defenders
Expanding computer security knowledge

Linux Kernel and Network Monitoring with BPF

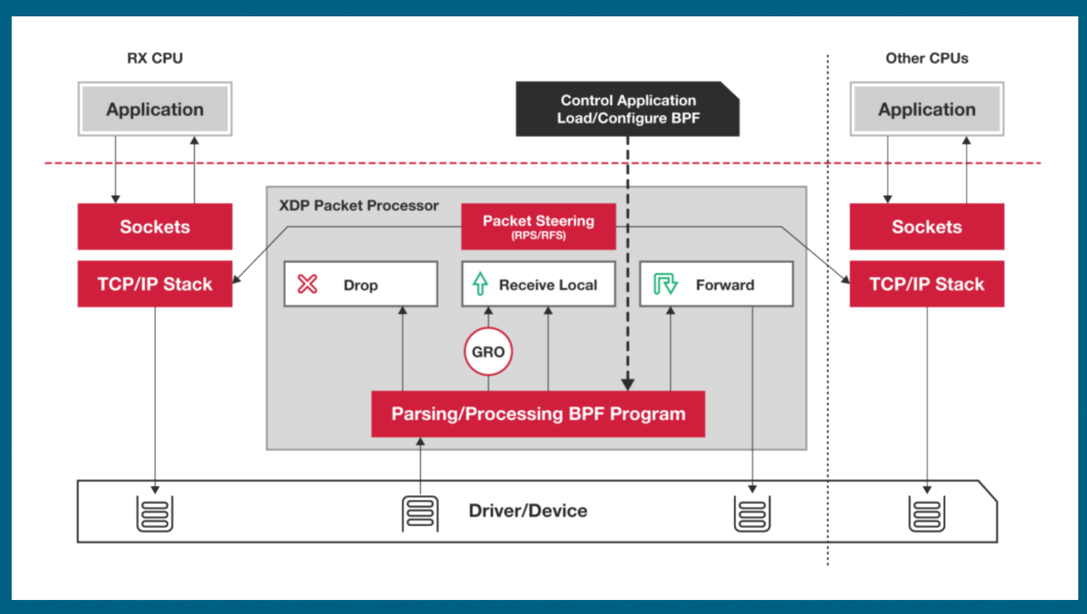
Elias Augusto, University of Maryland College Park

Dominic Genna, University of Texas at Austin

Project Mentors: Tim Toole 5631, Rick Strong 5632

Problem Statement:

- The Extended Berkley package filter (eBPF) is a powerful tool for network and performance monitoring at the kernel level. How can eBPF's capabilities be utilized to develop active monitoring, tracing, and live debugging capabilities for the Linux kernel?
- The eXpress Data Path (XDP), a feature of the Linux kernel, allows users to supply an eBPF program to monitor and operate on inbound network traffic early in the network stack. We want to understand how the filtering and packet direction changing capabilities of XDP programs are better compared to preexisting solutions.



1. Where XDP fits in before the Linux network stack

Objectives and Approach:

- The current objective related to Linux kernel instrumentation is to utilize eBPF to detect and gain information about other eBPF programs, from the information that they pass to non-kernel programs to the kernel functions and features that they target. We are currently pursuing this information by monitoring eBPF-related Linux syscalls and parsing information based on code analysis of the Linux kernel.
- The goal for the XDP work is to determine the extent to which we can modify network packets destinations by using the XDP actions DROP and REDIRECT. This is being tested by creating different virtual machine configurations and placing the XDP program on network interfaces within these virtual machines. We can then monitor the network traffic on each of the virtual machines to see how the traffic is getting processed and routed.

Results

- By monitoring the BPF syscall, we were able to capture custom eBPF programs, parse information from eBPF programs related to the Linux service manager, and dump bytecode describing the functionality of these programs from memory.

```
Task: ringbut_submit. PID: 1740 Msg: Creating Map
Map Type: BPF_MAP_TYPE_RINGBUF
Task: ringbut_submit. PID: 1740 Msg: Key Size:
Task: ringbut_submit. PID: 1740 Msg: Value Size: 0
Task: ringbut_submit. PID: 1740 Msg: Max Entries: 65536
Task: ringbut_submit. PID: 1740 Msg: End Entry


Task: ringbut_submit. PID: 1740 Msg: Loading program
Program Type: BPF_PROG_TYPE_TRACEPOINT
Instruction buffer:
 1:     bf16000000000000
 2:     1811000003000000
 3:     0000000000000000
 4:     b70200004c000000
 5:     b703000000000000
 6:     8500000083000000
 7:     bf07000000000000
 8:     b700000001000000
 9:     15070e0000000000
 10:    7963180000000000
 11:    bf71000000000000
 12:    b702000040000000
 13:    8500000072000000
 14:    7961100000000000
 15:    6317400000000000
 16:    7961200000000000
 17:    6317440000000000
 18:    6961280000000000
 19:    6317480000000000
 20:    bf71000000000000
 21:    b702000000000000
 22:    8500000084000000
 23:    b700000000000000
 24:    9500000000000000
Task: ringbut_submit. PID: 1740 Msg: Verifier Log Verbosity: 0
Task: ringbut_submit. PID: 1740 Msg: Verifier Log:
Task: ringbut_submit. PID: 1740 Msg: Instruction Count: 24
Task: ringbut_submit. PID: 1740 Msg: Allocated instruction space
Task: ringbut_submit. PID: 1740 Msg: End Entry
```

- Created an eBPF program that allows a user to define sets of networking rules based on protocols, addresses and ports, which are then loaded onto the network interface the user chooses. This program also has many different output modes, which display different statistics about current network traffic and what actions are being taken on that traffic.

```
XDP_DROP:        24
XDP_PASS:        1966
XDP_TX:          0
XDP_REDIRECT:    0

TCP Packets:     1964
UDP Packets:     18
ICMP Packets:    8
```

Impact and Benefits:

- Syscall parsing and code analysis efforts will allow for both the development of eBPF related kernel monitoring capabilities and a deeper understanding of eBPF's underlying functionalities.
- Using XDP for processing network traffic allows for less computing resources to be utilized for enforcing basic network rules, and has quicker response times to possible network related threats.

1. https://www.iovisor.org/technology/xdp

Sandia National Laboratories

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC., a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525. SAND2017-XXXXC.

U.S. DEPARTMENT OF ENERGY
National Nuclear Security Administration

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.