

SharkFest'26 Europe

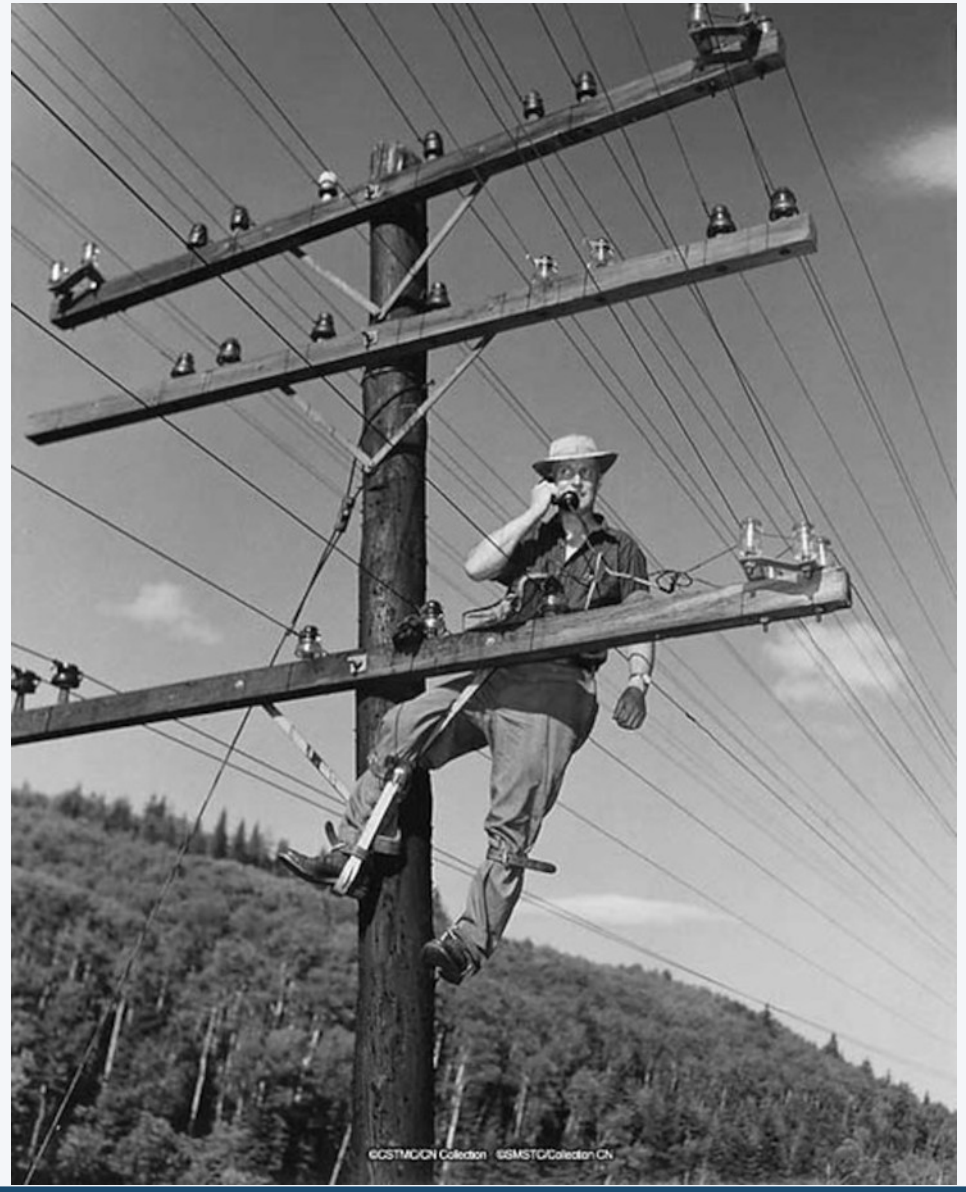
Packet capture basics; how to get 'em

Jaap Keuter

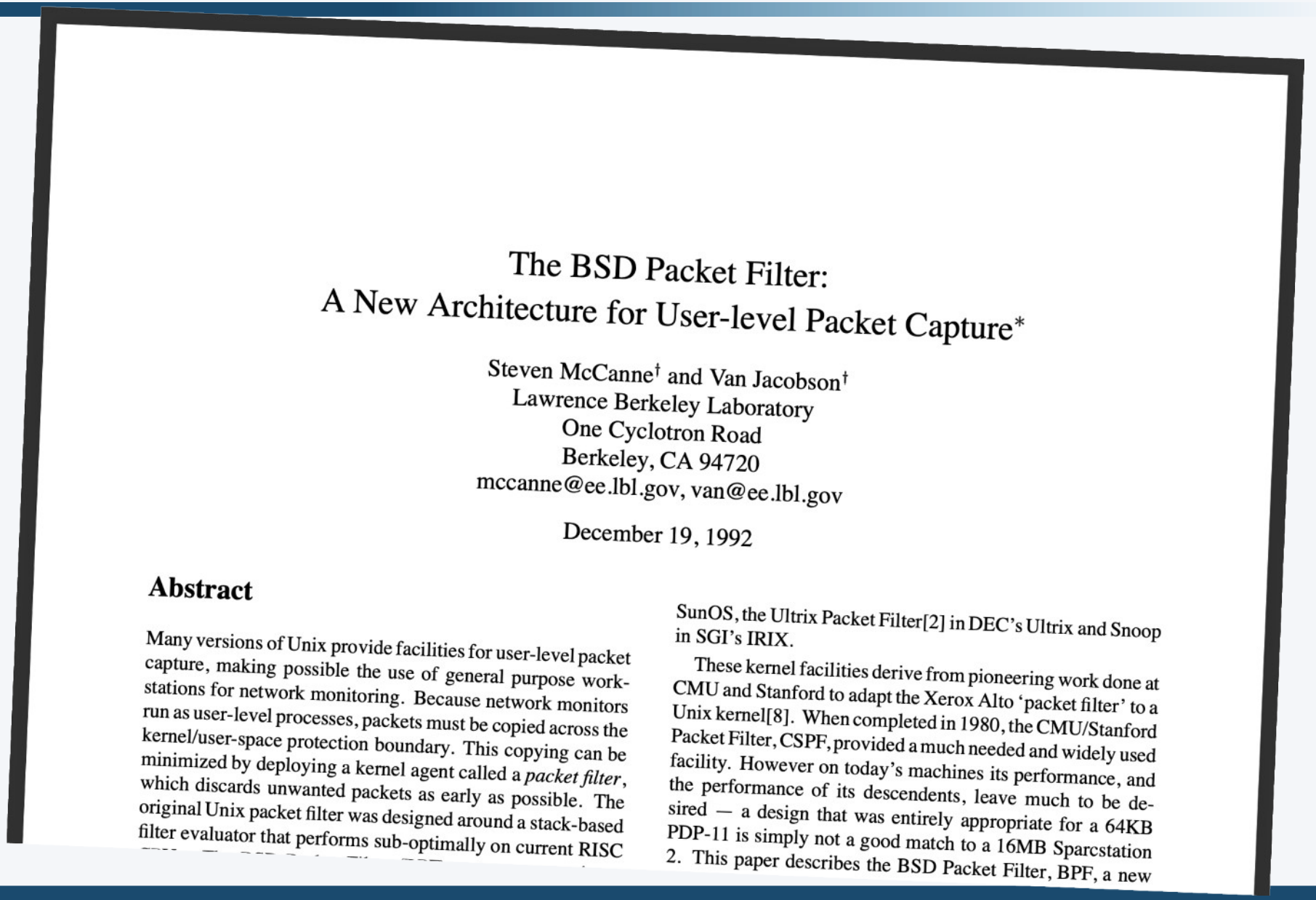
WIRESHARK

Sponsored by:
PROFITAP DEEP NETWORK OBSERVABILITY
cPacket
TCP/IP training
endace Record. Respond.

An age-old story: Monitoring



BSD Packet Filter (BPF)



The BSD Packet Filter:
A New Architecture for User-level Packet Capture*

Steven McCanne† and Van Jacobson†
Lawrence Berkeley Laboratory
One Cyclotron Road
Berkeley, CA 94720
mccanne@ee.lbl.gov, van@ee.lbl.gov

December 19, 1992

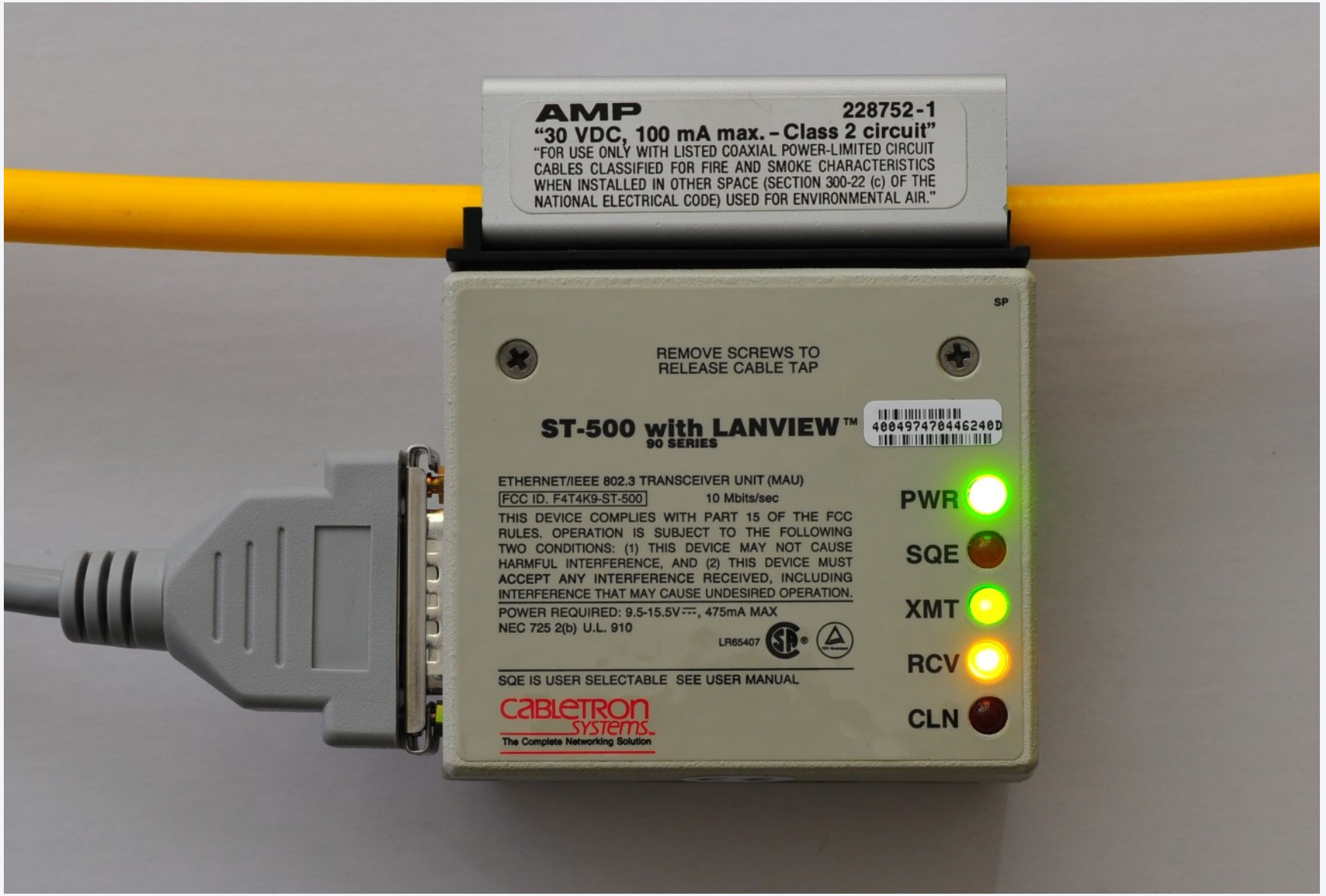
Abstract

Many versions of Unix provide facilities for user-level packet capture, making possible the use of general purpose workstations for network monitoring. Because network monitors run as user-level processes, packets must be copied across the kernel/user-space protection boundary. This copying can be minimized by deploying a kernel agent called a *packet filter*, which discards unwanted packets as early as possible. The original Unix packet filter was designed around a stack-based filter evaluator that performs sub-optimally on current RISC

SunOS, the Ultrix Packet Filter[2] in DEC's Ultrix and Snoop in SGI's IRIX.

These kernel facilities derive from pioneering work done at CMU and Stanford to adapt the Xerox Alto 'packet filter' to a Unix kernel[8]. When completed in 1980, the CMU/Stanford Packet Filter, CSPF, provided a much needed and widely used facility. However on today's machines its performance, and the performance of its descendents, leave much to be desired — a design that was entirely appropriate for a 64KB PDP-11 is simply not a good match to a 16MB Sparcstation 2. This paper describes the BSD Packet Filter, BPF, a new

Vampire clamp on bus topology



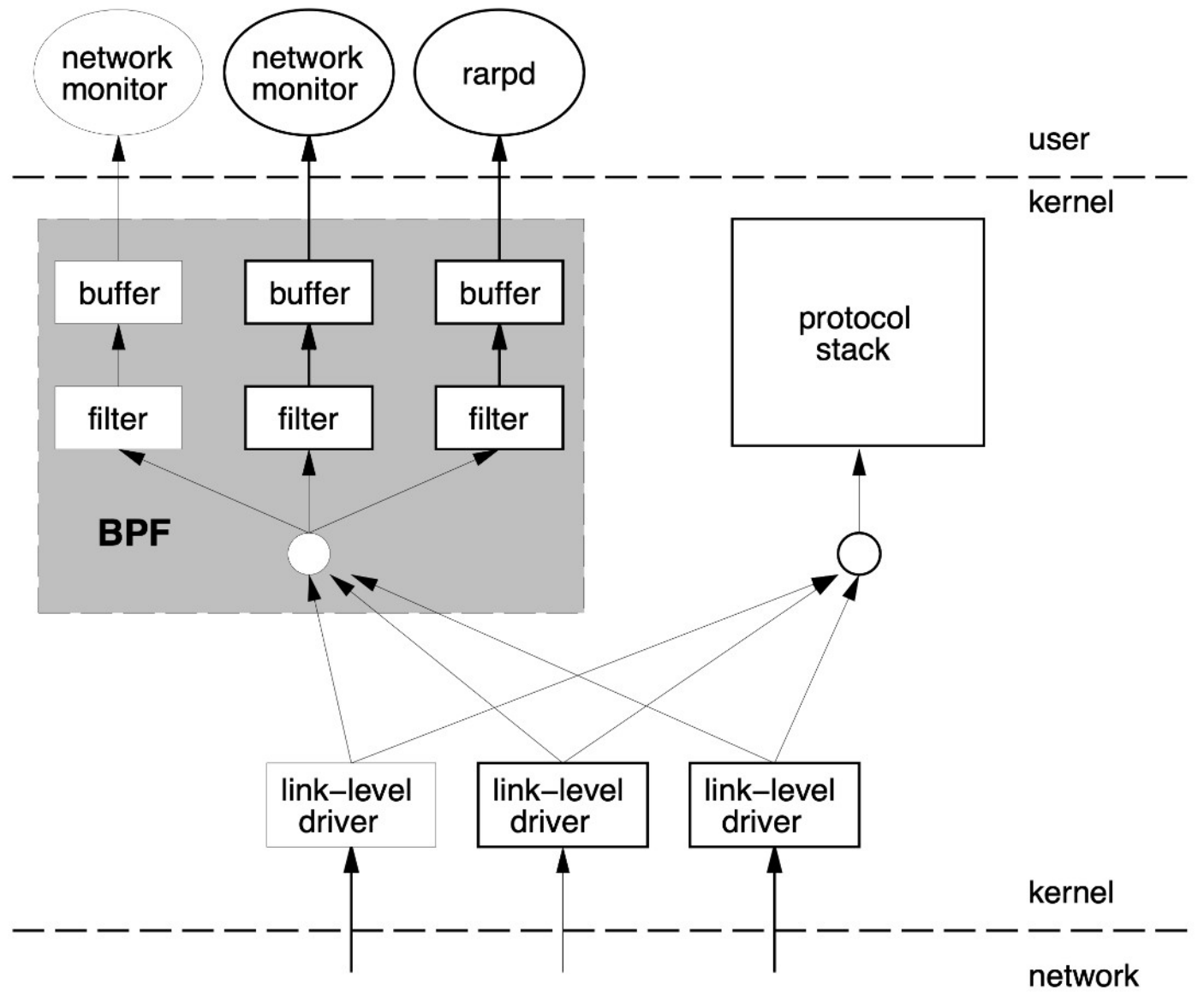
Van Jacobson (SharkFest ‘10)



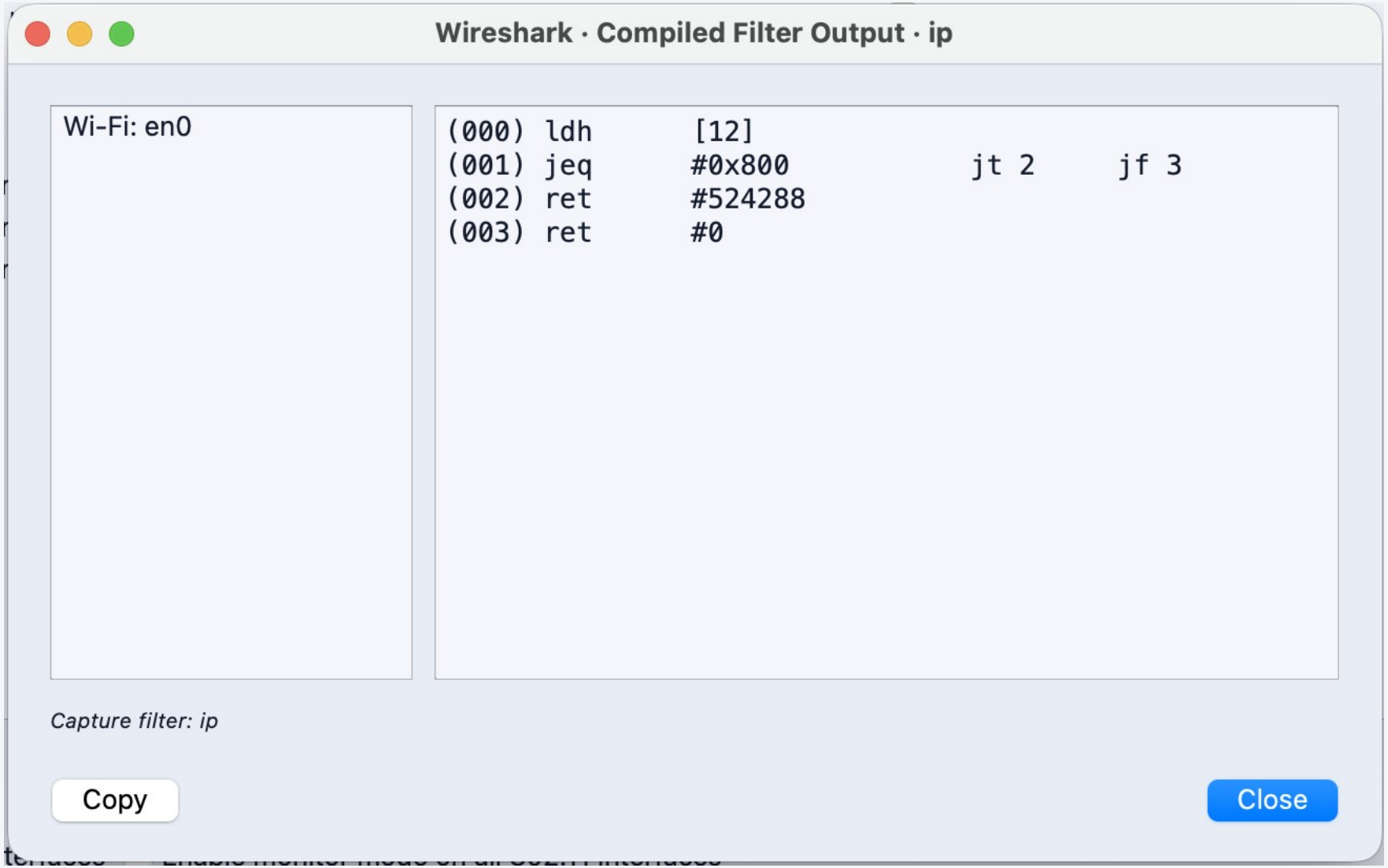
Steve McCanne (SharkFest ‘11)



BPF architecture



Wireshark capture filter → BPF



BPF the language

HOME | SECURITY | FAQ | MAN PAGES | CI | LINK-LAYER HEADER TYPES | SEE ALSO | OLD RELEASES | THANKS!

TCPDUMP & LIBPCAP

PCAP-FILTER(7) MAN PAGE

Updated: 12 September 2026 • View in plain text • Index • Return to Main Contents

This man page documents libpcap version 1.11.0 (see also: git master branch, 1.10.7, 1.10.6, 1.10.5, 1.10.4, 1.10.2, 1.10.1, 1.10.0, 1.9.1, 1.8.1, 1.7.4, 1.6.2, 1.5.3).

Your system may have a different version installed, possibly with some local modifications. To achieve the best results, please make sure this version of this man page suits your needs. If necessary, try to look for a different version on this web site or in the man pages available in your installation.

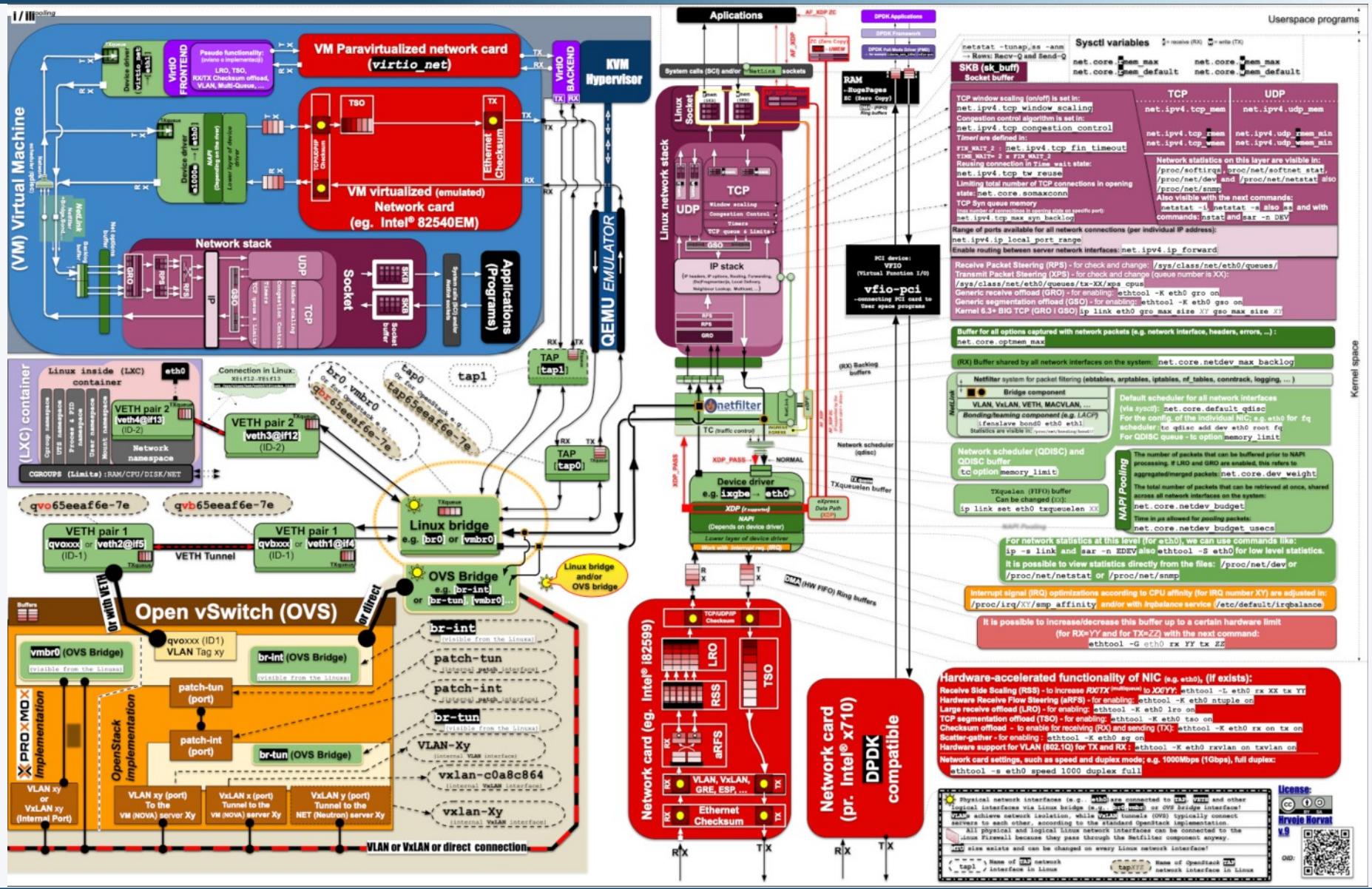
NAME

pcap-filter - packet filter syntax

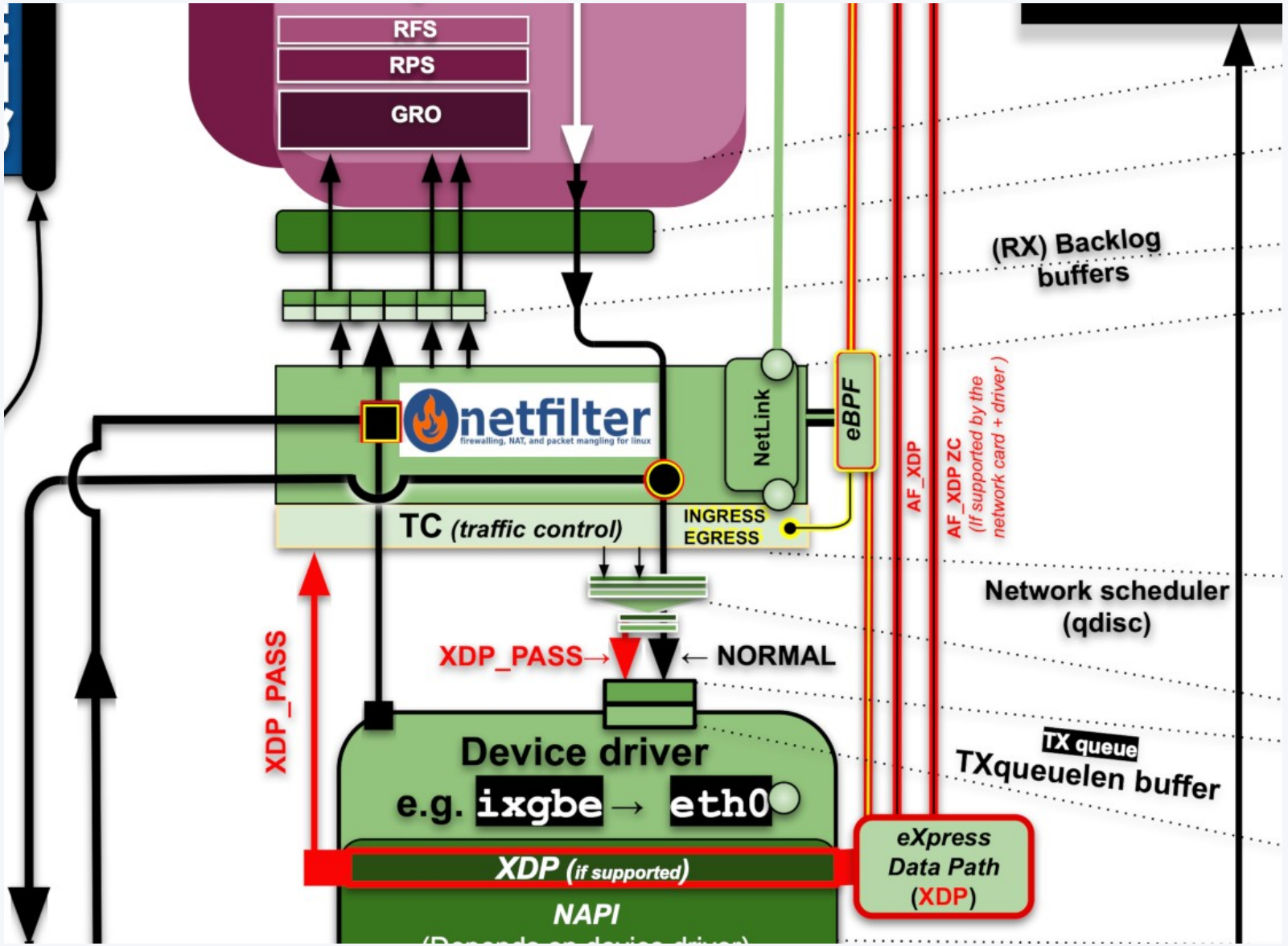
DESCRIPTION

pcap_compile(3PCAP) is used to compile a string into a filter program. The resulting filter program can then be applied to some stream of packets to determine which packets will be supplied to **pcap_loop**(3PCAP), **pcap_dispatch**(3PCAP), **pcap_next**(3PCAP), or **pcap_next_ex**(3PCAP).

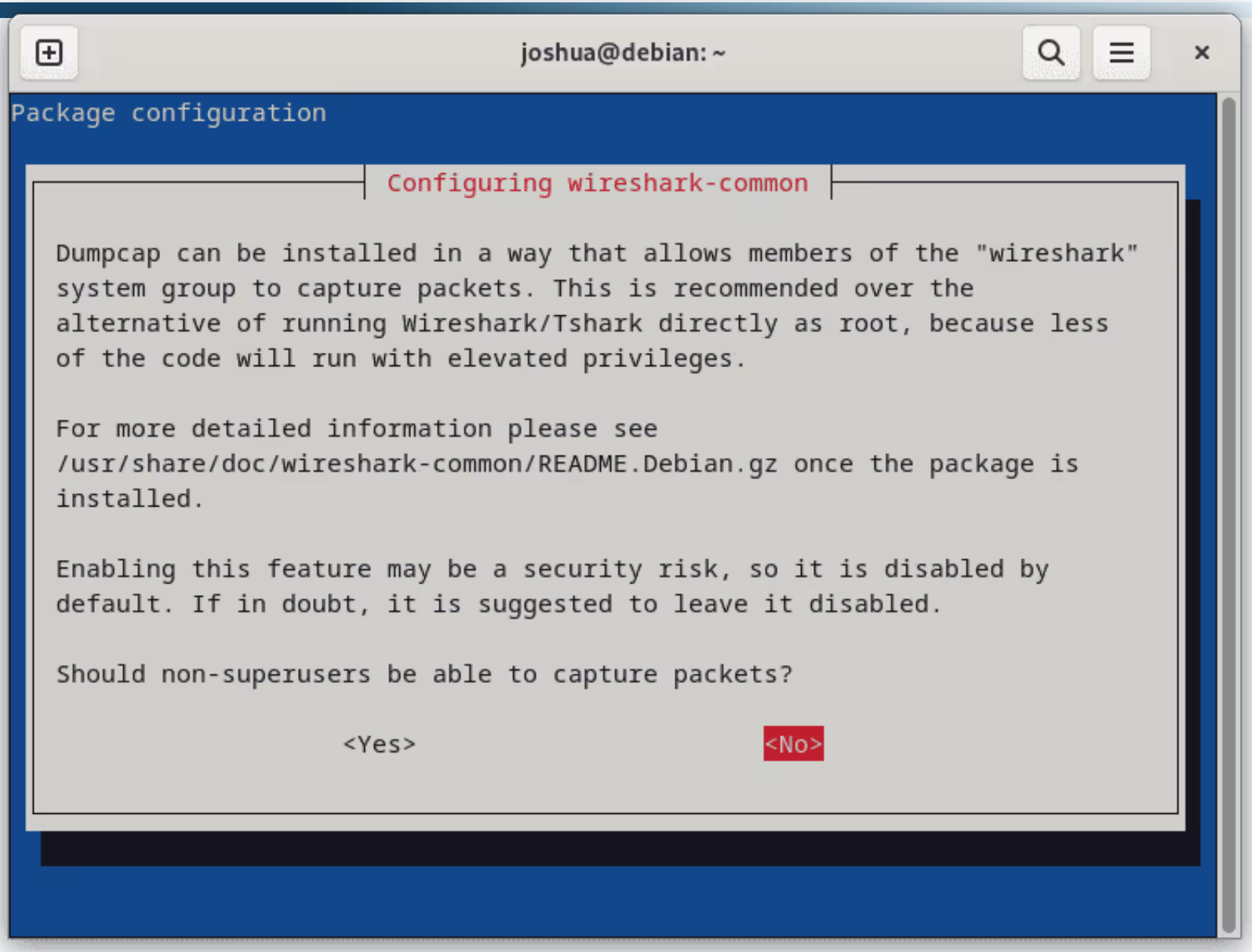
Linux kernel network infrastructure



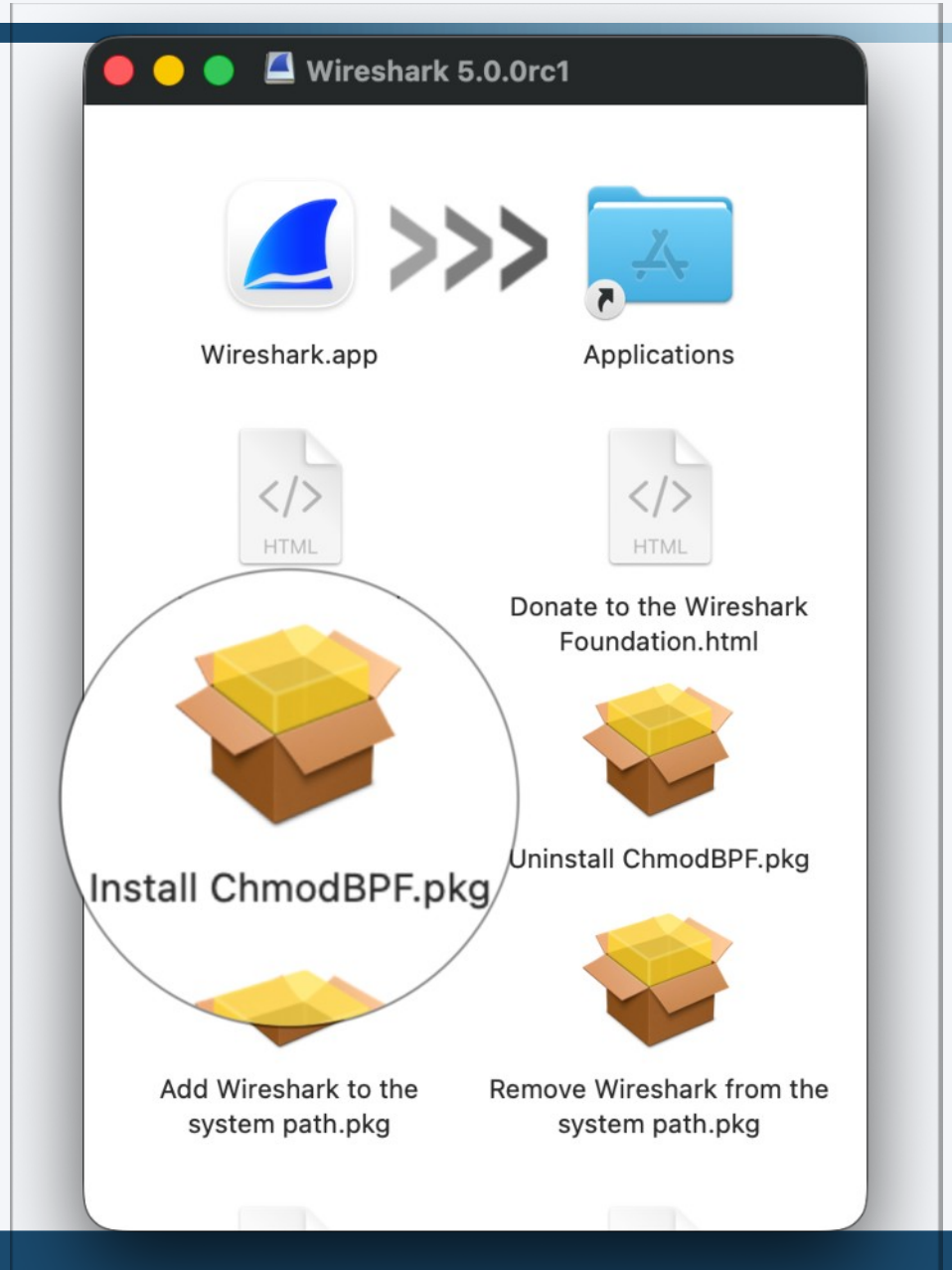
Linux kernel - eBPF



Permissions – Debian GNU/Linux



Permissions - MacOS



WinPcap

An Architecture for High Performance Network Analysis

Fulvio Risso and Loris Degioanni
Dipartimento di Automatica e Informatica – Politecnico di Torino
Corso Duca degli Abruzzi, 24 – 10129 Torino, Italy
{risso,degioanni}@polito.it

Abstract *— Most Unix systems provide a set of system calls that allow applications to interact with the network directly. These primitives are useful for example in packet capture applications, which need to grab the data flowing through the network without any further processing from the kernel. WinPcap is a newly proposed architecture that adds these functionalities to Win32 operating systems. WinPcap includes a set of innovative features (such as packet monitoring and packet injection) that are not available in previous systems. This paper*

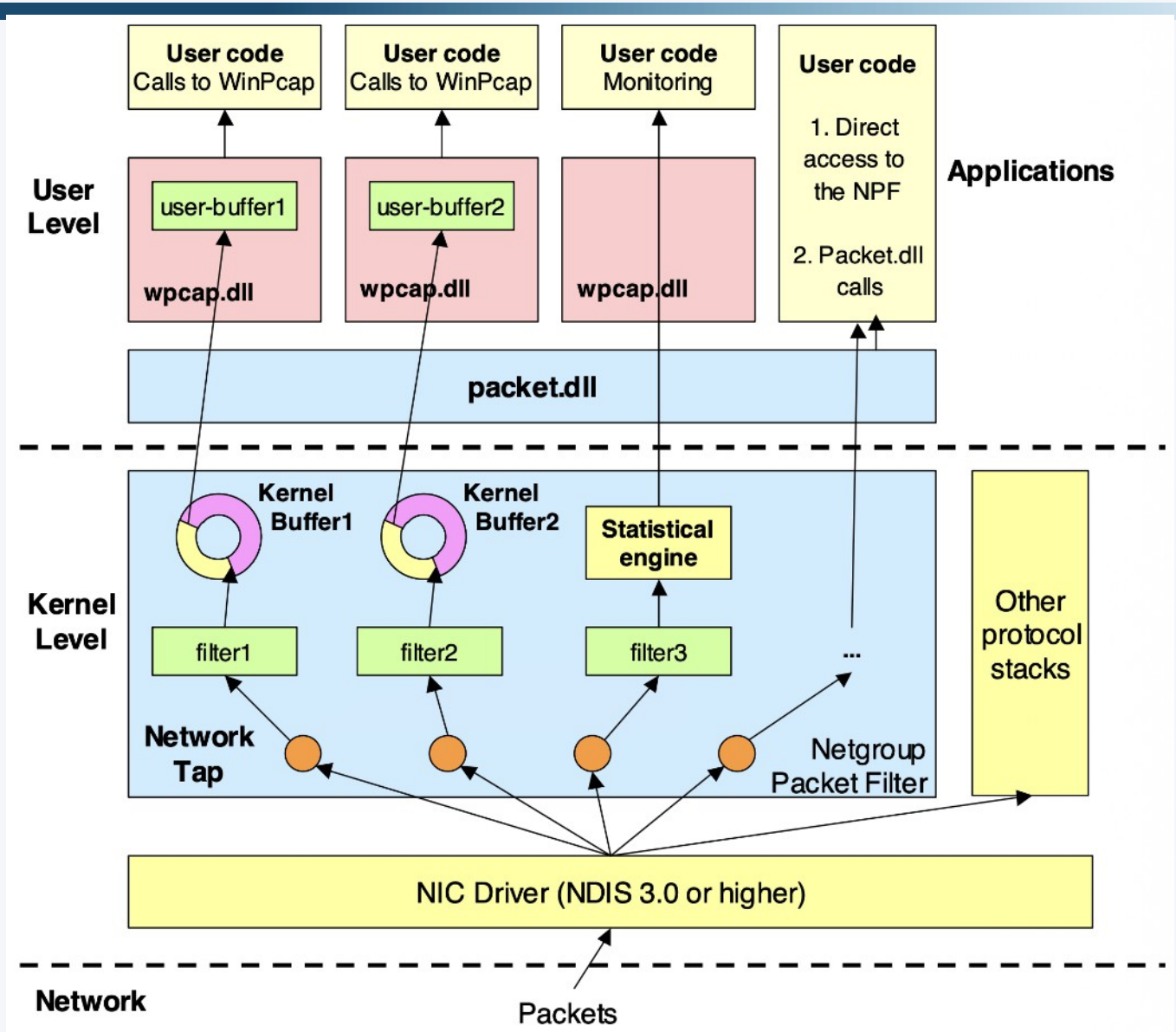
capture on Win32 and it fills an important gap between Unix and Windows. Furthermore WinPcap puts performance at the first place, thus it is able to support the most demanding applications.

Packet filtering is made up of a kernel-mode component (to select packets) and a user-mode library (to deliver them to the applications). Last component provides a standard interface for low-level network access and allows programmers to avoid kernel-level programming. WinPcap includes an optimized kernel-

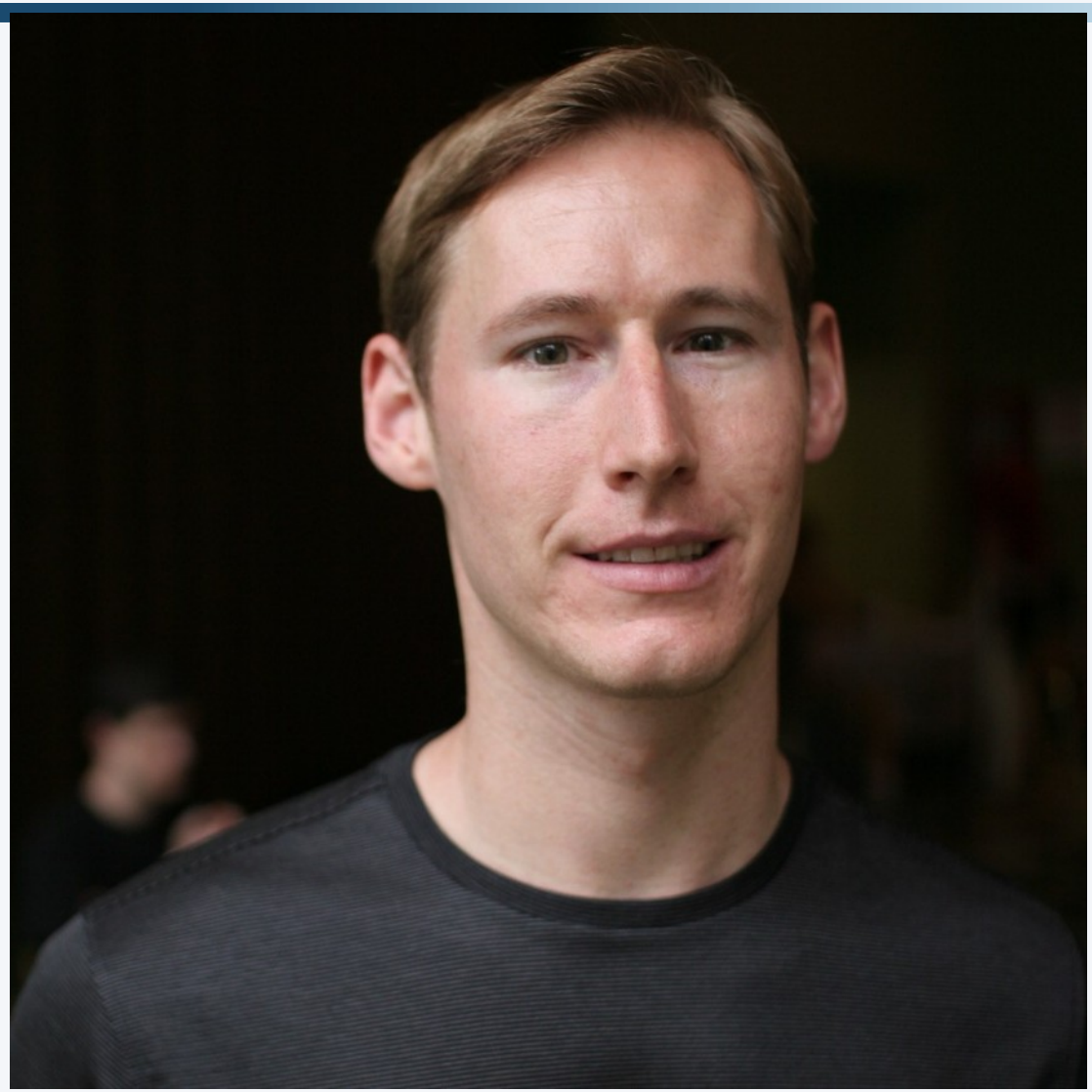
Loris Degioanni



WinPcap architecture

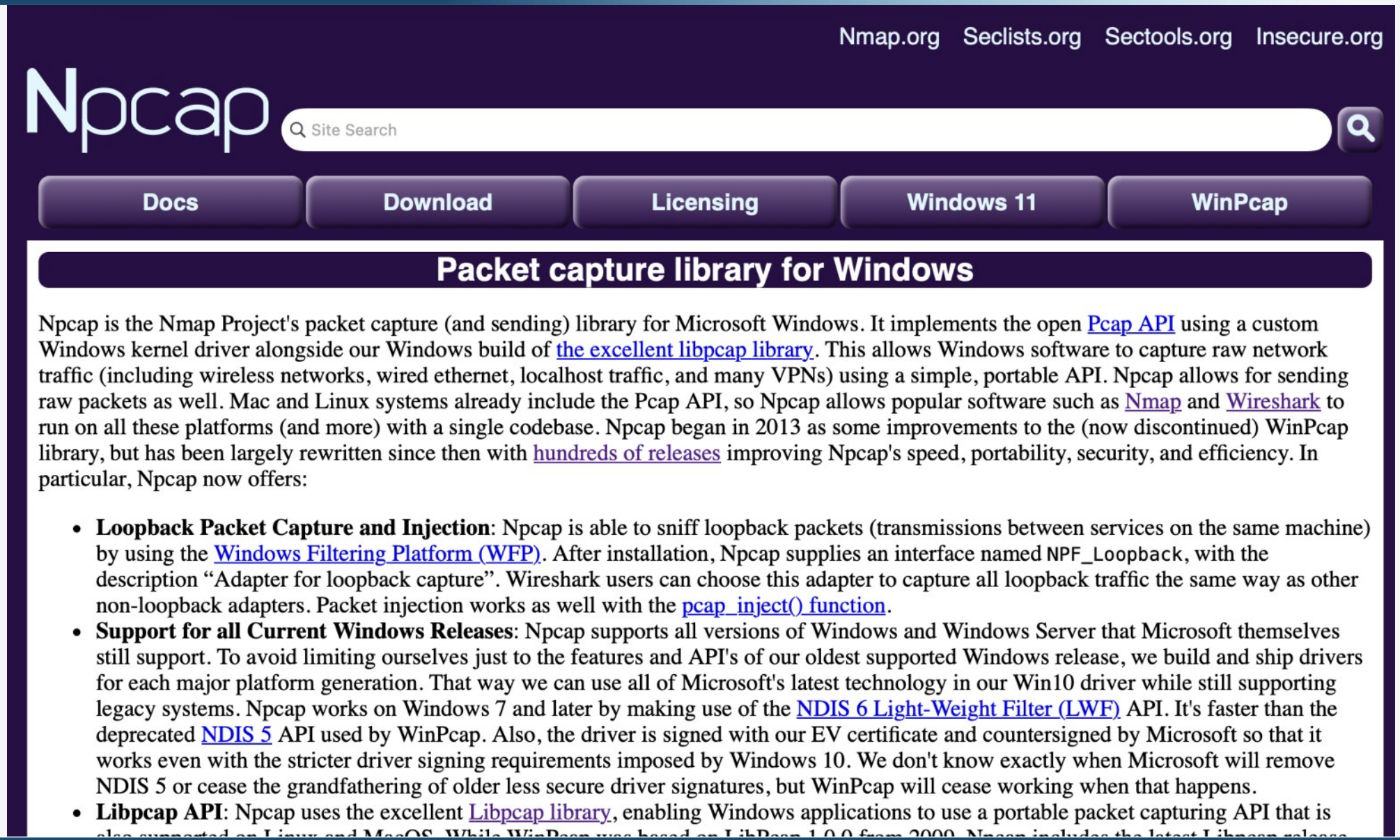


Gordon Lyon (aka. Fyodor)



By Jacob Appelbaum - https://www.flickr.com/photos/ioerror/196447899/, CC BY-SA 2.0, https://commons.wikimedia.org/w/index.php?curid=1258996

Npcap, by Nmap



Nmap.org Seclists.org Sectools.org Insecure.org

Npcap

Site Search

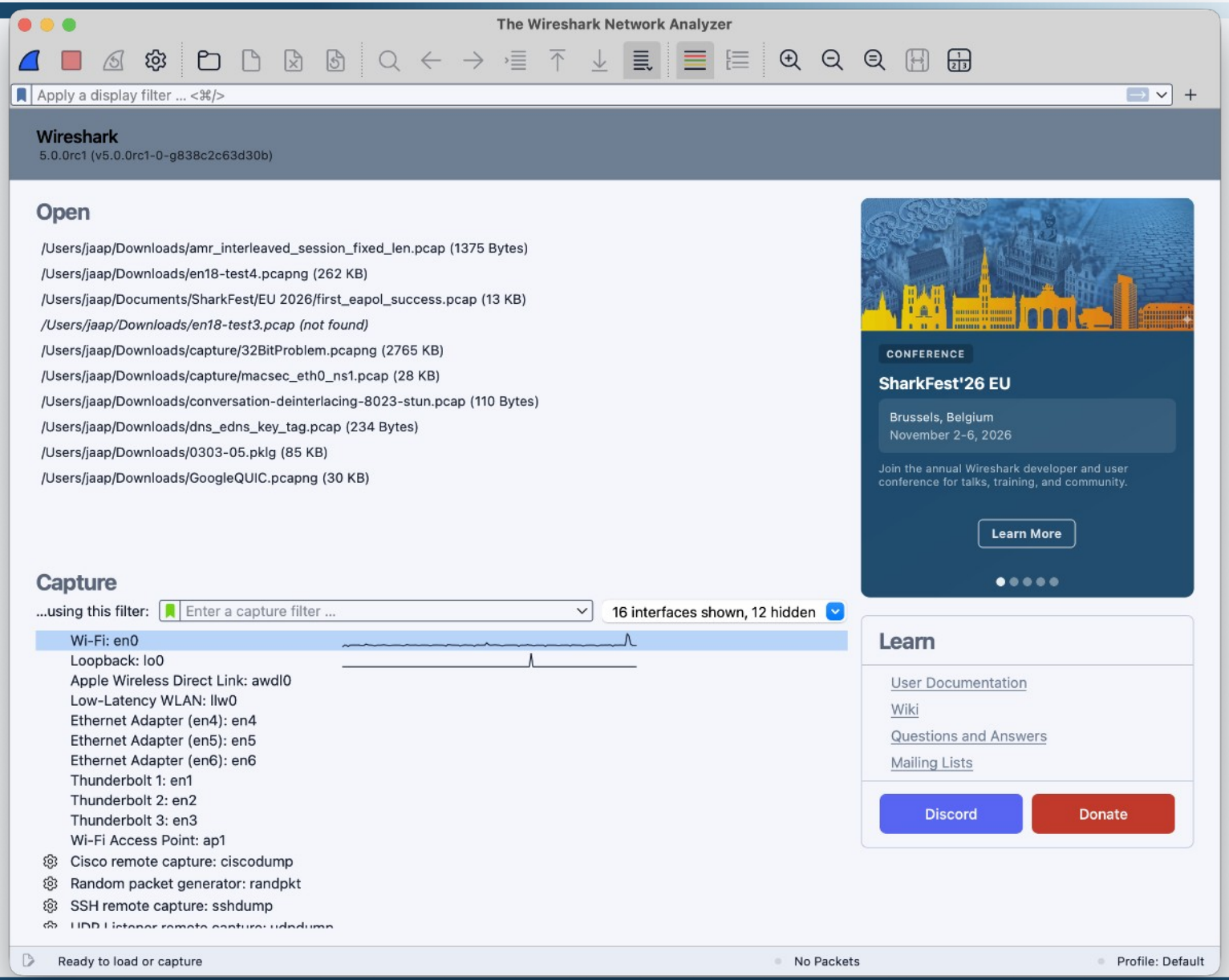
Docs | Download | Licensing | Windows 11 | WinPcap

Packet capture library for Windows

Npcap is the Nmap Project's packet capture (and sending) library for Microsoft Windows. It implements the open Pcap API using a custom Windows kernel driver alongside our Windows build of the excellent libpcap library. This allows Windows software to capture raw network traffic (including wireless networks, wired ethernet, localhost traffic, and many VPNs) using a simple, portable API. Npcap allows for sending raw packets as well. Mac and Linux systems already include the Pcap API, so Npcap allows popular software such as Nmap and Wireshark to run on all these platforms (and more) with a single codebase. Npcap began in 2013 as some improvements to the (now discontinued) WinPcap library, but has been largely rewritten since then with hundreds of releases improving Npcap's speed, portability, security, and efficiency. In particular, Npcap now offers:

- **Loopback Packet Capture and Injection**: Npcap is able to sniff loopback packets (transmissions between services on the same machine) by using the Windows Filtering Platform (WFP). After installation, Npcap supplies an interface named `NPF_Loopback`, with the description "Adapter for loopback capture". Wireshark users can choose this adapter to capture all loopback traffic the same way as other non-loopback adapters. Packet injection works as well with the pcap_inject() function.
- **Support for all Current Windows Releases**: Npcap supports all versions of Windows and Windows Server that Microsoft themselves still support. To avoid limiting ourselves just to the features and API's of our oldest supported Windows release, we build and ship drivers for each major platform generation. That way we can use all of Microsoft's latest technology in our Win10 driver while still supporting legacy systems. Npcap works on Windows 7 and later by making use of the NDIS 6 Light-Weight Filter (LWF) API. It's faster than the deprecated NDIS 5 API used by WinPcap. Also, the driver is signed with our EV certificate and countersigned by Microsoft so that it works even with the stricter driver signing requirements imposed by Windows 10. We don't know exactly when Microsoft will remove NDIS 5 or cease the grandfathering of older less secure driver signatures, but WinPcap will cease working when that happens.
- **Libpcap API**: Npcap uses the excellent Libpcap library, enabling Windows applications to use a portable packet capturing API that is

Wireshark welcome screen

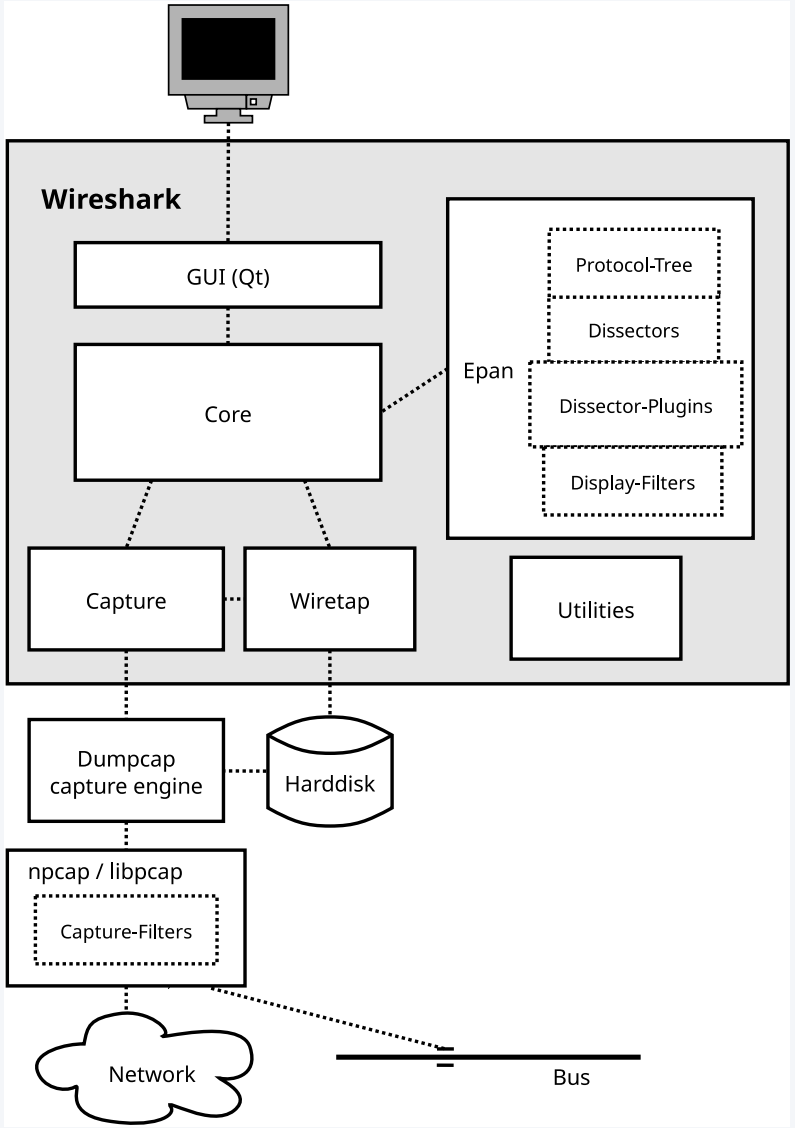


Packet capture, what else



Does it capture or does is not?

- Does Wireshark capture packets ?
  - Yes
  - No
- The answer is: No
- Dumpcap is about 0.5% of the size of Wireshark, therefore much less code that has to have elevated privileges to be able to capture.



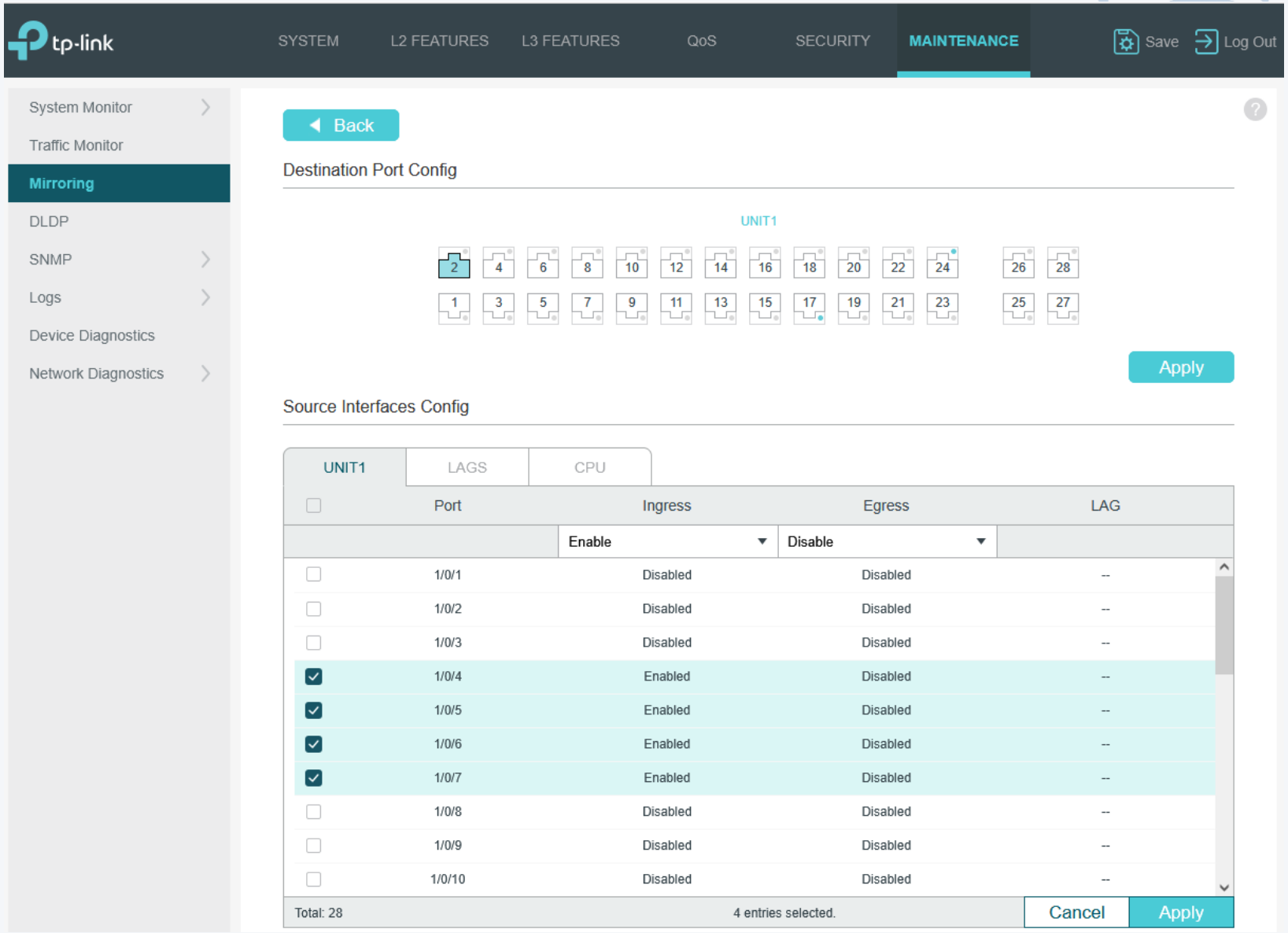
Long term capture

- What's best to use for long term capture?
  - Wireshark
  - Dumpcap

- The answer is: Dumpcap
- Dumpcap is only collects packets in (multiple) files, while Wireshark does analysis. This consumes memory, which eventually runs out.
- There do exist other options out their, which may even be more suitable for your needs.

How to capture “the rest of the network”?

- A mirror or SPAN port, often available on managed switches, allows you to receive a copy of the packets on selected ports. You than only have to capture on that mirror port.
- This may even be remote, e.g. on Cisco devices through the use of ERSPAN, which tunnels the traffic to you.
- Be aware of the capacity restrictions though.

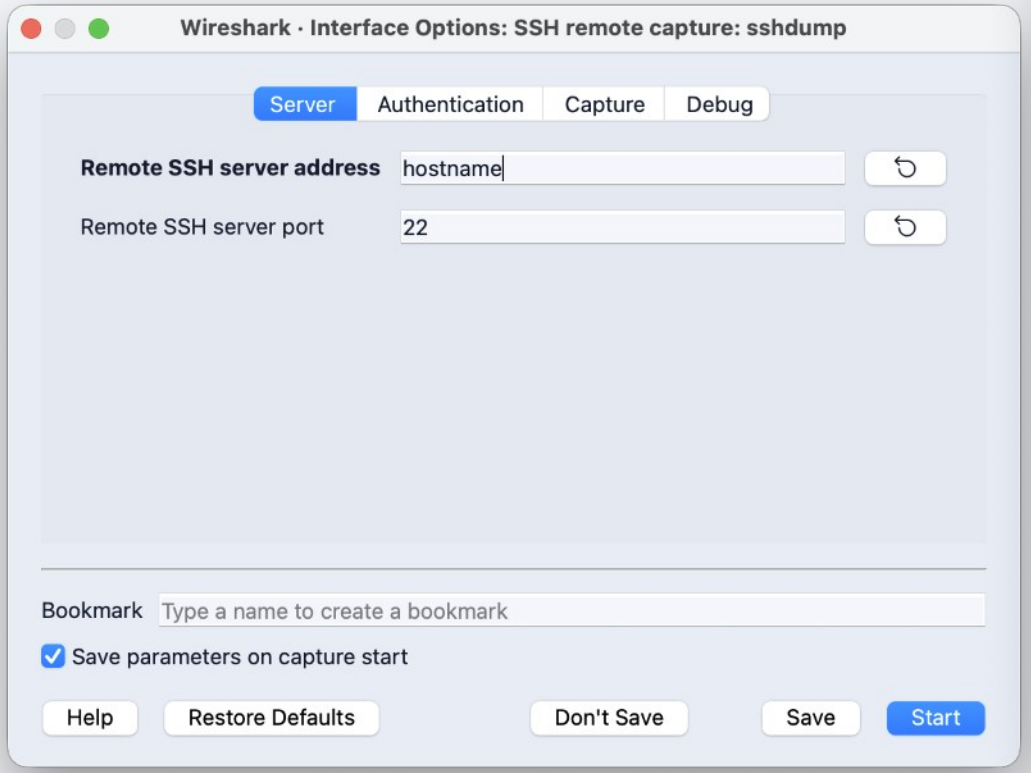


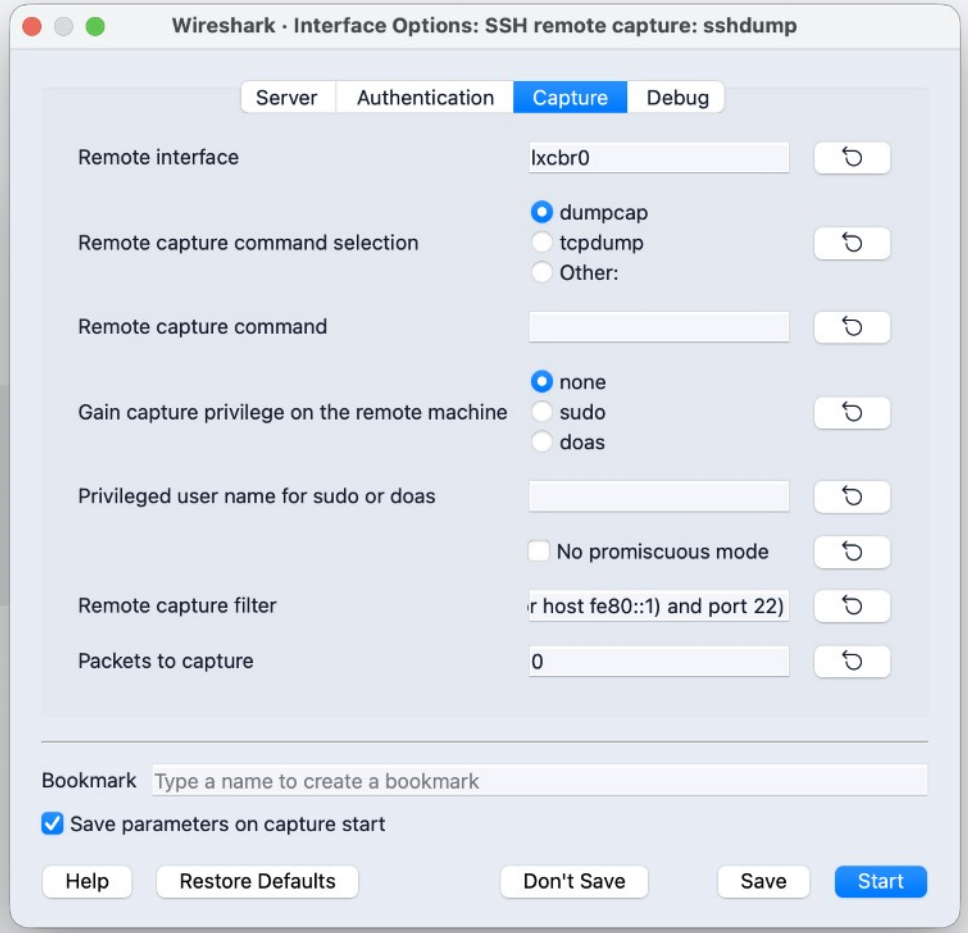
How to capture “the rest of the network”?

- For higher capacity capture options you’ll have to revert to dedicated hardware and software solutions.
  - This you can build it yourself, i.e. using solutions from ntop.org
  - Go look at the vendors here, i.e. Endace, Profitap and cPacket.

Capture from other sources?

- Through the extcap interface any program can be a capture device.
  As long as it produces packets.
- Wireshark comes with a few installed, but vendors make their own, e.g.,
  - Profitap capture plugin for ProfiShark
  - Siemens csharg extcap for Edgeshark





How about WiFi capture?

- Answer: It depends…
- In the early days (IEEE 802.11b, a and g) things were simple, when the RF chipset and driver supported it.
  There's a constant push/pull between RTL chipsets and MediaTek.
- Note that these cards need to run in monitor mode, to allow full RF control. Depending on the chipset capabilities and driver quality, this may or may not allow for simultanious association.

  Monitor mode is not to be confused with promiscuous mode, which applies to the Ethernet level, after RF demodulation
- These days (IEEE 802.11n and beyond) MIMO and beamforming come into play, making over-the-air interception really hard.
  The next best thing is the capture from the AP's themselves, which better ones now support.
- Have a look at the Kismet website (kismetwireless.net) for way more information on this.

Thank You!

WIRESHARK

https://conference.wireshark.org/sf26eu/talk/7ELU8V/feedback/

