PacketLab - Tools Alpha Release and Demo

Tzu-Bin Yan*, Yuxuan Chen*, Anthea Chen*, Zesen Zhang†,

Bradley Huffaker†‡, Ricky Mok†‡, Kirill Levchenko*, kc claffy†‡

*University of Illinois at Urbana-Champaign †UC San Diego ‡CAIDA
{tbyan2,yuxuanc5,anxuec2,klevchen}@illinois.edu,zez003@eng.ucsd.edu,{bradley,cskpmok,kc}@caida.org

ABSTRACT

The PacketLab universal measurement endpoint interface design facilitates vantage point sharing among experimenters and measurement endpoint operators [1]. We have continued working on fleshing out the design details of PacketLab components and adding enhancements to facilitate adoption. These include designing the PacketLab certificate system, adding support for measurement creation via a wrapper tool and a C library module, enhancement of reference endpoint ability for measurement flexibility and experiment scheduling, and devising a proxy program to accommodate experimenters without a public IP address. With the code base stabilizing, we are ready to announce our first open release of the PacketLab software package (available at `pktlab.github.io`). We invite network measurement researchers to try out our tools and welcome any feedback from the research community.

CCS CONCEPTS

• **Networks** → **Network measurement**;

ACM Reference Format:
Tzu-Bin Yan*, Yuxuan Chen*, Anthea Chen*, Zesen Zhang†, Bradley Huffaker†‡, Ricky Mok†‡, Kirill Levchenko*, kc claffy†‡. 2022. PacketLab - Tools Alpha Release and Demo. In *Proceedings of the 22nd ACM Internet Measurement Conference (IMC '22), October 25–27, 2022, Nice, France.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3517745.3563029

1 INTRODUCTION

Due to the cost-prohibitive nature of maintaining large distributed measurement infrastructure, network measurement researchers have relied on sharing measurement infrastructure for proper vantage point access. Such behavior introduces difficulties:

- **Compatibility**: Varying platform designs require the taxing process of porting experiments across platforms.
- **Incentives**: Deploying new experiments requires platform operator support.
- **Trust**: Platforms supporting custom measurements must limit access to vetted researchers for security reasons.

IMC '22, October 25–27, 2022, Nice, France

ACM ISBN 978-1-4503-9259-4/22/10…$15.00
https://doi.org/10.1145/3517745.3563029

PacketLab [1] addresses the aforementioned problems through two unique design decisions: (1) exporting the **network interface** of endpoints to experimenters via a common mechanism - the PacketLab protocol, and (2) including the use of **cryptographic certificates** with **program-based restrictions** for endpoint access control. Exporting a common interface helps experiment compatibility via the notion of *write once, run anywhere*. Moreover, exporting a common interface reduces platform operator burden for new experiments as no effort other than interface maintenance is required. The cryptographic certificate design enables a scalable trust architecture as operators can use PacketLab experiment privilege provisioning capabilities to customize restrictions.

2 DESIGN OVERVIEW

The PacketLab architecture consists of measurement endpoints, brokers (or rendezvous servers), and experiment controllers. We summarize the overall design next.

Measurement endpoints and experiment controllers. Measurement endpoints in PacketLab are software or hardware agents having access to the Internet. Endpoints are run by endpoint operators, representing parties owning and volunteering network resources. The network interface of the endpoint is exported via the PacketLab protocol, which supports BSD-socket-API-like requests. Experiment controllers, operated by experimenters, issue protocol requests to endpoints to carry out measurements. Though resembling the BSD socket API, there are two core differences: (1) Sending packets in PacketLab allows specifying the target send time. This allows execution of network measurements requiring precise timing and enables experimenters to avoid access link contention during measurement runtime by scheduling traffic. (2) Endpoints support buffering of received packets to prevent interference with measurement results. To facilitate the process of endpoint controller match-making, PacketLab introduces brokers that support experiment subscription/publication, as well as subscription/publication matching and endpoint experiment notification.

Access control. PacketLab performs coarse-grained access control using cryptographic certificates. Endpoint operators, when allowing experimenter access to endpoints, can do so by signing an experiment-privilege certificate and sending the certificate to the experimenter. An experimenter can delegate privileges by signing and sending a delegate-privilege certificate to another experimenter. PacketLab *agents* (endpoints, controllers, and brokers), during operations such as endpoint-controller TLS handshake and broker

experiment subscription/publication, perform coarse-grained access control based on whether the remote-party-submitted certificate chain represents valid privilege for the operation. PacketLab also supports finer-grained restrictions on supplied privileges using eBPF-based monitor and filter programs. Endpoints apply filters and monitors at experiment runtime against received network packets and protocol messages to determine if a packet/message is retrievable by controllers and whether to permit a request. Certificate signers at signing time determine the set of filters and monitors to be applied by endpoints by including filter/monitor program hashes within the signed certificate. The endpoint, before protocol request serving, requests the programs from the controller and enforces them afterward during runtime. To ease certificate signers' burden on finer-grained privilege restriction specifications, signers can also choose to include predefined constraints such as experiment priority restrictions and maximum experiment run time within the signed certificates. The endpoint enforces such predefined constraints during experiment setup and runtime as well.

3 CURRENT RESULTS

In the past few years, we have fleshed out design details and introduced enhancements to the PacketLab toolset. We achieved:

Certificate Design. To provide support for privilege manipulation among PacketLab entities, we introduced the use of custom X.509 certificates for privilege provisioning and delegation. We chose X.509 certificates as they allow us to (1) use existing well-established tools and libraries for certificate generation and parsing, as well as (2) incorporate certificate chain checking during TLS handshakes. To facilitate credential management, we devised utilities (the `pktlab` Python module with the `PPKSMan` command-line tool) for credential generation and management.

Support for Measurement Creation. To support diverse measurements, we created a wrapper tool `pktwrap` for re-purposing existing libc-based network applications to use PacketLab for network communication without modification to applications. Because experimenters are more used to writing measurement programs against the OS BSD socket interface, within the C utility library `libpktlab`, we introduced a new module `pktif`, which provides a BSD-socket-like API to facilitate the writing of PacketLab measurements.

Enhancement of Reference Endpoint Ability. To increase measurement flexibility (e.g. allowing arbitrary TCP traffic without inducing kernel responses), we added support to the reference endpoint program for raw socket kernel view traffic-*stealing* using a Linux loadable kernel module. We also added preliminary experiment scheduling support for the reference endpoint program via a numbering priority system. When publishing an experiment, the controller can specify a target experiment execution priority. The endpoint, when notified, performs experiment scheduling based on the supplied priority to allow the known-highest-priority experiment to run and queue up all lower-priority experiments.

Accommodation of Experimenters without Public IP. As experimenters may not own public IP addresses for controllers to accept incoming endpoint connections, we designed and implemented a proxy program `pktproxy` that enables volunteers to serve as an intermediate rendezvous points for controllers and endpoints.

Open Release for Software Package. With our code base stabilizing, we are ready to announce our first open release of the PacketLab software package. Our package includes a standalone reference endpoint implementation (`pktendpt`) and an experiment manager utility program (`pktxpmgr`) for easy experiment publication and sample measurement execution. Our software package currently comes with sample measurements for DNS A record lookup, HTTP GET request issuing, as well as ICMP echo request sending and reply receiving. With a publicly accessible broker server at CAIDA (`pktbrokr.caida.org`), external researchers can now try out PacketLab firsthand. To introduce the community to our tools, we plan to provide in-person demonstrations during the IMC poster session. Our plan is to provide a setup where conference participants can request experiment privilege certificates (potentially through a web interface) for CAIDA-controlled endpoints and run sample measurements on the endpoints using their own laptops. For interested readers, our package can be found at `pktlab.github.io`.

4 FUTURE ROADMAP

We will continue our work to develop PacketLab components and enhancements including support for monitor programs on `pktendpt` (currently under development) and evaluating and improving `pktendpt` performance based on further testing with various measurements. We also plan to (1) develop off-the-shelf measurement applets for common measurements such as ping, traceroute, general HTTP requests, general DNS record lookup, and TLS certificate retrieval, (2) cooperate with external parties to support PacketLab on their measurement platform, (3) complete a formal verification of the PacketLab access control design, (4) investigate further experiment scheduling options for `pktendpt`, (5) explore potential mechanisms to reward endpoint operators for exporting more endpoint capabilities, and (6) document the full PacketLab design. We hope by carefully-documenting our design and providing the community with a reference PacketLab endpoint implementation, we can facilitate the process for established measurement platforms to add support for the PacketLab interface. As we envision PacketLab to become an open source project supported by the measurement community in the future, we also plan to publicize our source code repository soon to accept community contributions.

5 CONCLUSION

Given our progress in design and development, we believe PacketLab is now ready for the measurement community. We invite measurement researchers to try out our tools and welcome any feedback for further improvements.

This work is based on research sponsored by the National Science Foundation (NSF) grants CNS-1764055 and CNS-1903612. The views herein are those of the authors and do not necessarily represent endorsements, either expressed or implied, of NSF.

REFERENCES

[1] Kirill Levchenko, Amogh Dhamdhere, Bradley Huffaker, kc claffy, Mark Allman, and Vern Paxson. 2017. Packetlab: A Universal Measurement Endpoint Interface. In *Proceedings of the 2017 Internet Measurement Conference (IMC '17)*. Association for Computing Machinery, 254–260.