PNA: A Passive Network (Monitoring) Appliance

Michael J. Schultz
Ben Wun Renault Young Patrick Crowley

Washington University in Saint Louis

October 1, 2010
`10 01 10`

Outline

1. Introduction
   - A Passive Network Appliance
   - Motivation & Background
2. Design & Implementation
   - Design Goals
   - Implementation
   - Evaluation
3. Conclusions
   - Summary
   - Future Work

What Was That?

Passive Network Appliance

Passive We only listen to traffic, we don’t create our own to monitor the network

Appliance We want to make this a low-cost, easy-to-install commodity item for network administrators

Motivation

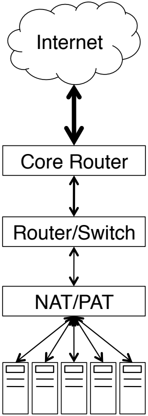
- Network Abuse—Worms (e.g. Conficker) and Scanners
  - Good attacks aren't noticed by the end-user
- Misconfiguration—Setting up a large network can be hard
  - Outcomes are not always as expected
  - Repercussions can be bad and diagnosis can be difficult

We want to help detect and diagnose "abnormal" behavior as quickly as possible [1]

[1] We define "abnormal" informally (i.e. not based on a model)

Background[2]

- Network-based Intrusion Detection and Prevention Systems (IDPS)
- Host-based IDPS
- Network Behavior Analysis (NBA)



[2] Based on NIST's "Guide to Intrusion Detection and Prevention Systems (IDPS)"

Process Every Packet

Why Know how all traffic flows through network

How Avoid using packet sampling

- Sampling selects packets to process and ignores others

Just read the protocol information and write to hash table

Don’t depend on packet contents

Why Encryption obscures any meaningful packet data

Deep Packet Inspection takes more time

How Just read ...

- ... source and destination IPs
- ... protocol
- ... source and destination ports

Track Statistics in Real-Time

Why Allows Real-Time monitors of traffic flows

How Store counts in hash tables for quick access

- Bytes, Packets
- Connections—Unique hosts a local device has talked to
- Sessions—Unique conversations between pair of hosts

Low-cost and Distributed

Why Enable “drilling down” into the network

- Behind network address (port) translators (NATs/PATs)

How Developed around Linux kernel

- Works on commodity systems, from servers to low-end PCs

Install anywhere PNA node is needed with minimal setup

Aggregate Data to One Place

Why Too hard to analyze data scattered about a network

How Move all the data into a central data store

- For example: Amazon's Simple Storage Service + Elastic Compute Cloud
- Enable global analysis of data
- Simplify off-line analysis of data

Future Work: Use the same software for both on- and off-line analysis!

Packet Processing I

- Packets are mirrored from switch to PNA device

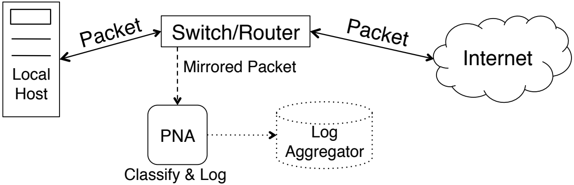


Figure: Abstract View of Network with PNA

- Built around Linux kernel (vanilla + `ip_promisc` patch)
- Uses “netfilter” API input hooks

Packet Processing II

1. Software grabs the header data (IPs, protocol, ports)
2. Inserts into three hash tables
    - “local” IP table
    - “remote” IP table
    - “port” table

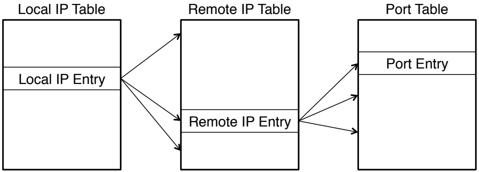


Figure: Three Level Hash Table Structure

3. Discards packet

Packet Processing II

1. Software grabs the header data (IPs, protocol, ports)
2. Inserts into three hash tables
   - “local” IP table
   - “remote” IP table
   - “port” table

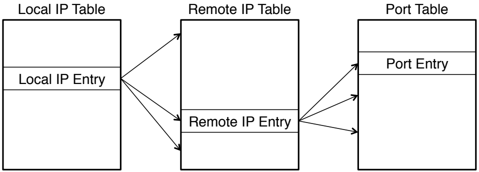


Figure: Three Level Hash Table Structure

3. Discards packet

Packet Processing II

1. Software grabs the header data (IPs, protocol, ports)
2. Inserts into three hash tables
   - “local” IP table
   - “remote” IP table
   - “port” table

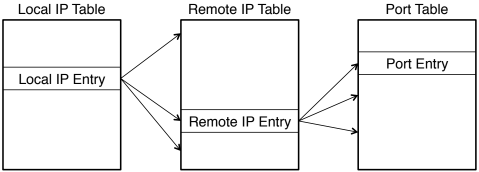


Figure: Three Level Hash Table Structure

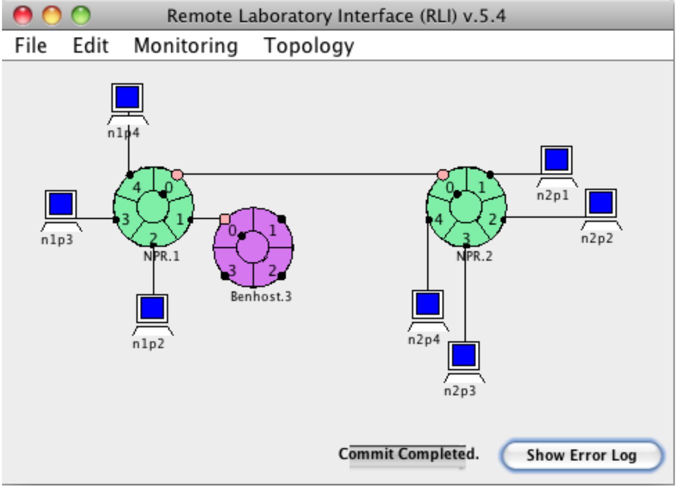
3. Discards packet

Track Statistics in Real-Time

- Use hash table structure to record
  - Per host—number of connections and sessions
  - Per host pair—number of packets, bytes, and ports
  - Per port pair—number of packets and bytes
- Every packet is checked against a threshold to find violators

First Demonstration

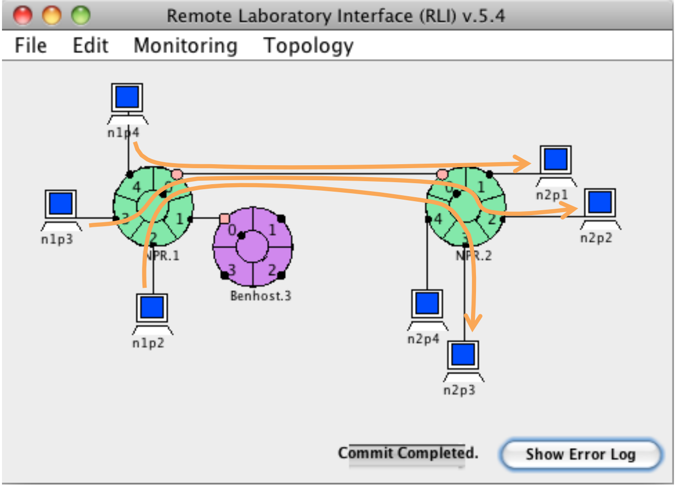
- Internal host becomes infected and starts scanning



Commit Completed.

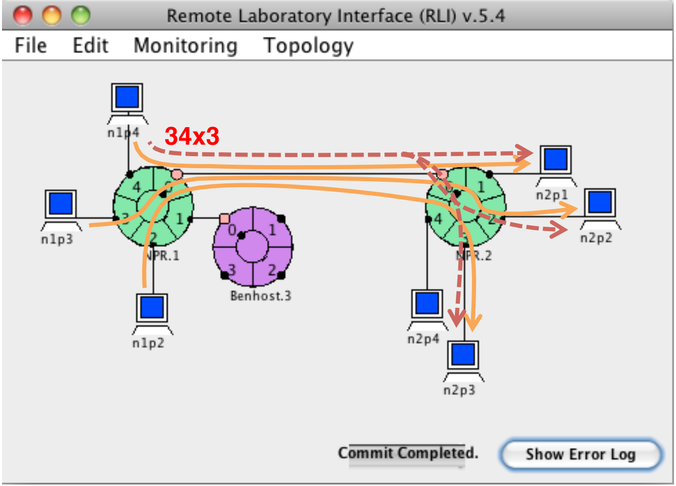
First Demonstration

- Internal host becomes infected and starts scanning



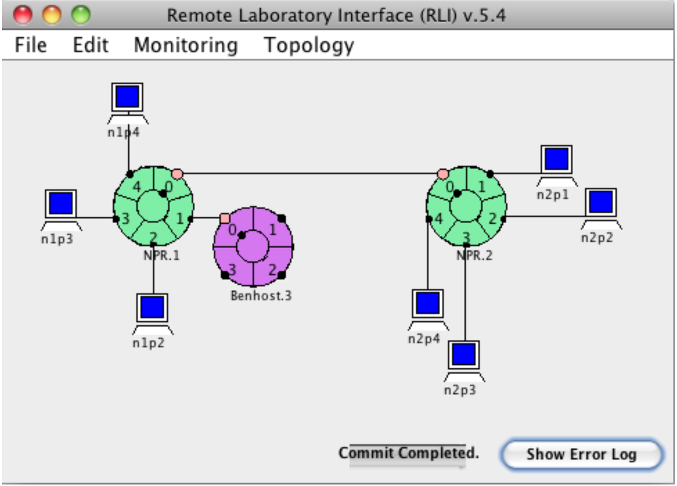
First Demonstration

- Internal host becomes infected and starts scanning



Second Demonstration

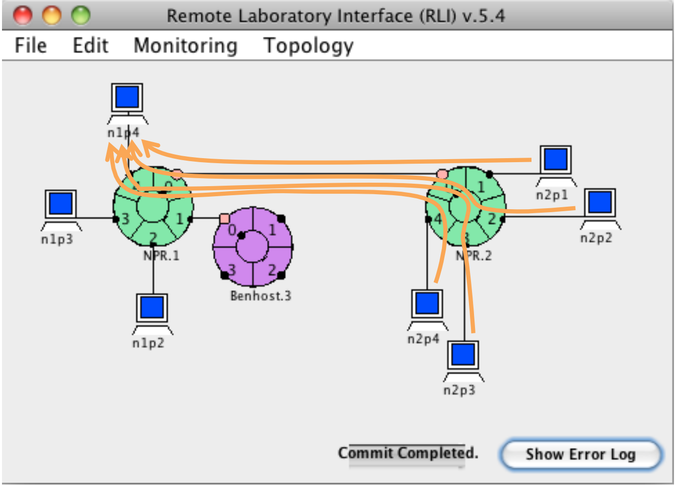
- External host starts attacking, fallback to whitelist



Commit Completed.

Second Demonstration

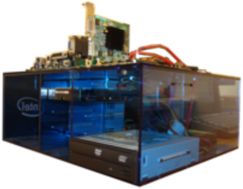
- External host starts attacking, fallback to whitelist



Commit Completed.

Low-cost and Distributed

- Use commodity systems
  - Low-end (Tolapai, Atom) to high-end servers
  - Compared to network equipment, high-end servers are cheap!



- Deploying a PNA node is “simple”
  - Add network cable and reconfigure the switch
- Archive data to central file store (takes some bandwidth)

We believe the costs of installation and logistics rival the cost of hardware.

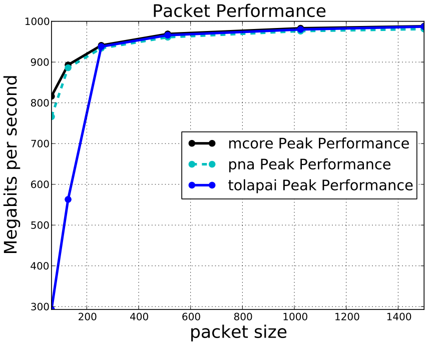
Aggregate Data

- Have used software to transfer off-line logs to S3
- Have built Hadoop MapReduce program to analyze logs off-line
- For privacy reasons, we expect network operator to initially resist sending *real* data to S3
    - For now, we archive data within the network

Performance Analysis

Graph of Throughput

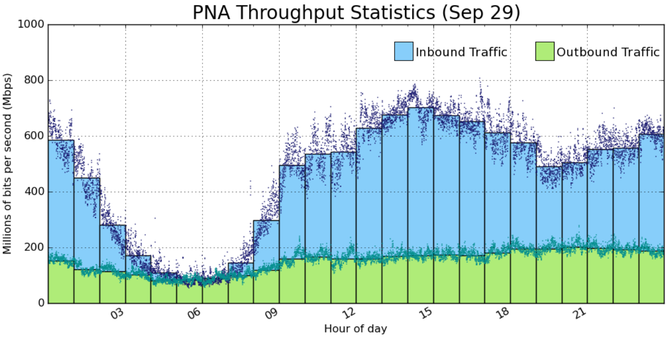
- "mcore" is eight-core 2.5 GHz with 16 GiB memory
- "pna" is eight-core 2.27 GHz with 12 GiB memory
- "tolapai" is single-core 1.4 GHz with 1 GiB memory



Performance Analysis

Graph of Live Throughput

- "pna" machine (8-core 2.27 GHz with 12 GiB memory)
- Deployed on NSS/NTS/IS&T Network



(Drops ∼ 1% of packets before our software sees it)

Summary

- Both internal and external network threats are hard to find
- Our PNA is able to detect and act on threats it sees, in real-time
- It performs fairly well under load in lab setting and in a live environment
- Has already detected semi-periodic scans of WUSTL network (that weren't seen before)

Future Work

- Continually Improve Software
  - Simplify event handling hooks
  - Increase throughput
- Deploy more PNA nodes in real networks
- Continue to work with network operators to make sure the tools and analyses are useful and effective
- Build up on-line and off-line analysis platform

Thanks for Listening