# Fight Fire with Fire

## s/Fire/Spoofing/g

Spoofing Detection in the UCSD Network Telescope

Raphael Hiesgen

INET, Hamburg University of Applied Sciences

CAIDA, UCSD

# IP Spoofing

- spoof, /sp$\overline{oo}$f/: hoax or trick (someone)
  - Trick someone into believing a packet was sent by someone else
  - *Problem:* No authentication in IPv4 headers (see IPSec AH)
- Reasons for spoofing
  - Conceal your “identity”
  - Impersonate someone else (MITM attack)
  - Denial of service (reflection attacks)

# Motivation

- Big problem throughout the Internet (e.g., DDoS)
- Our focus: impact on measurements
  - Research and operations depend on reliable data
  - Source address often used for geolocation
- Application domain: UCSD Network Telescope

# The UCSD Network Telescope

- A /8 darknet hosted at UCSD and operated by CAIDA
  - Hundreds of TB in *Internet Background Radiation* (IBR) per year
  - IBR examples: scans, malware, backscatter, …
    - One way traffic (unlike most communication on the Internet)
- Lots of research opportunities!
  - CSE student wrote her phd thesis on telescope measurements[1]
  - We will come back here later

[1] Leveraging Internet Background Radiation for Opportunistic Network Analysis, *Benson et al.*, IMC'15

# Data in Operational Use at IODA: Internet Outage Detection & Analysis

ioda.caida.org

BETA ioda Dashboard Alert Feed Project Info Acknowledgements Help Sign In caida

*Select a time range:*

September 5, 2016 12:00am UTC - September 12, 2016 12:00am UTC

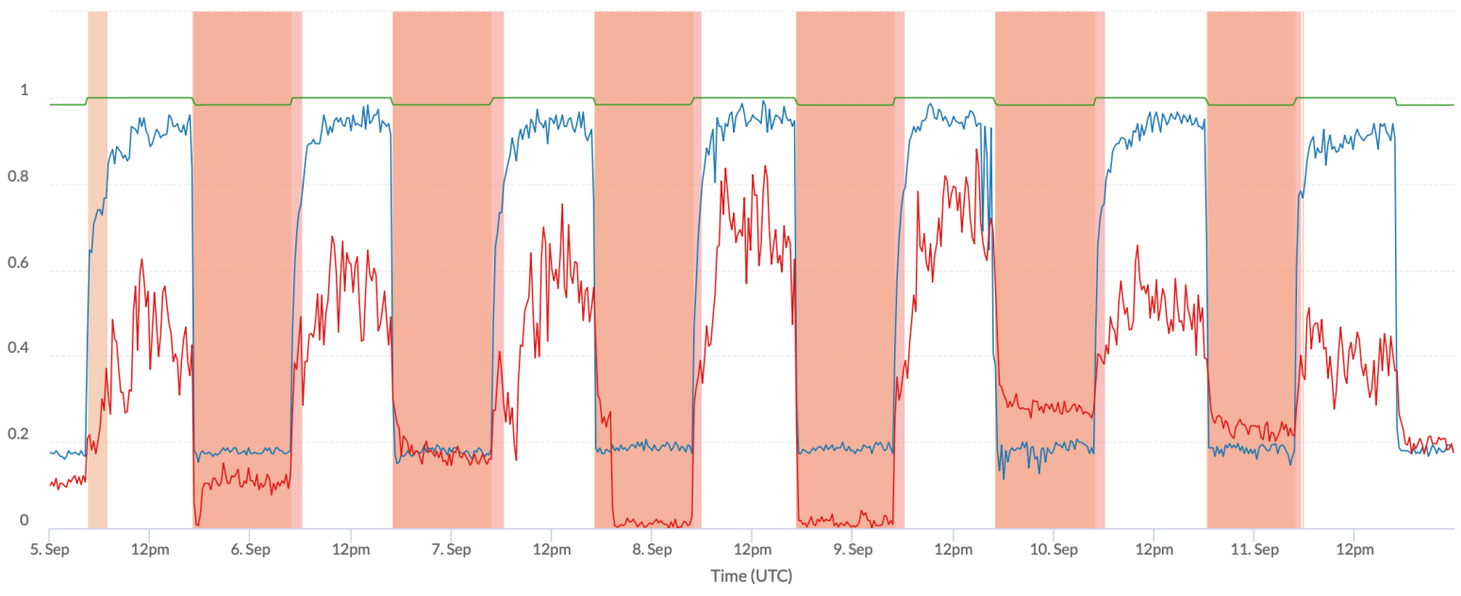
**Gabon**

**IODA Signals for Gabon**

Normalized Alert Bands

XY Graph

*Time point aggregation:* Avg of 4 pts/px (*20 minutes*)




# Series: 3 | # Points: 5040 | Data resolution: *5 minutes, 10 minutes*

*September 5, 2016 12:00am - September 12, 2016 12:00am*

**Alert Feed for Gabon**

*Show* 15 *entries* *Search:*

| | Time (UTC) | Type | Actual Value | Threshold |
|---|---|---|---|---|
| ✖ | Sep 11th 2016 5:00pm | BGP | 67,211 | 68,380 |
| ✖ | Sep 11th 2016 5:00pm | Active Probing | 41 | 146 |
| ✔ | Sep 11th 2016 6:00am | Active Probing | 120 | 146 |
| ✖ | Sep 11th 2016 5:50am | Active Probing | 115 | 146 |
| ✔ | Sep 11th 2016 5:40am | Active Probing | 119 | 146 |
| ✔ | Sep 11th 2016 4:55am | BGP | 68,380 | 68,380 |
| ✖ | Sep 10th 2016 6:30pm | Active Probing | 28 | 145 |
| ✖ | Sep 10th 2016 6:25pm | BGP | 67,211 | 68,380 |
| ✖ | Sep 10th 2016 6:20pm | Active Probing | 100 | 145 |

*Showing 1 to 15 of 35 entries* Previous Next

**Regional Outages for Gabon**

**Outages of ASes operating in Gabon**

https://ioda.caida.org/

# Our Goal

- Identify spoofed traffic in the IBR
- Challenges
  - One-way communication
  - Real-time processing
- No need to check every single packet

# Spoofing Detection

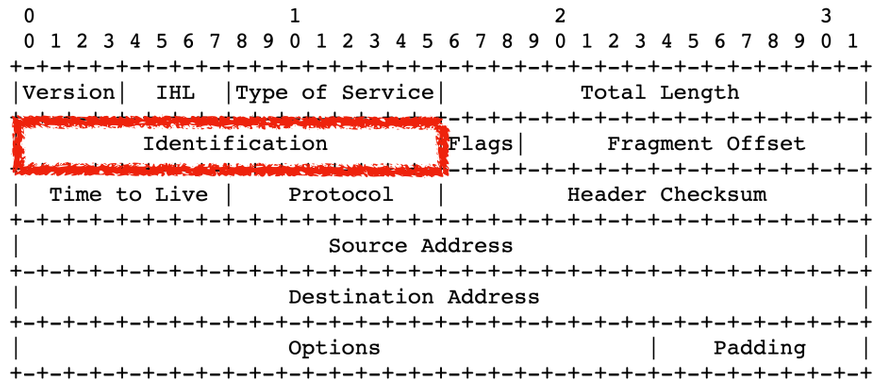
- Filter packets leaving your LAN
- Ingress and Egress filtering (RFC 2827 & 3704)
  - Whitelisting based on expected source addresses
  - Filters at IXPs based on customer cones and BGP[1]
  - Heuristics and rules[2]
    - Bursts of traffic including private and un-routed addresses
    - Packet anomalies (e.g., address ends in 0 or 255)

[1] Detection, Classification, and Analysis of Inter-Domain Traffic with Spoofed Source IP Addresses, *Lichtblau et al.*, IMC'17

[2] Estimating Internet address space usage through passive measurements, *Dainotti et al.*, CCR'14

# IP “Identification” Field

- 16 bits used to group fragments (RFC 791)
- Dubbed “IP ID”
- Traditionally a system-wide counter
  - Can be used to attribute packets to the same host
- First published by Steven M. Bellovin in 2002[1]
- Previous used at CAIDA for alias-resolution[2]




[1] A Technique for Counting NATted Hosts, *S. M. Bellovin*, Workshop of Internet Measurements ’02

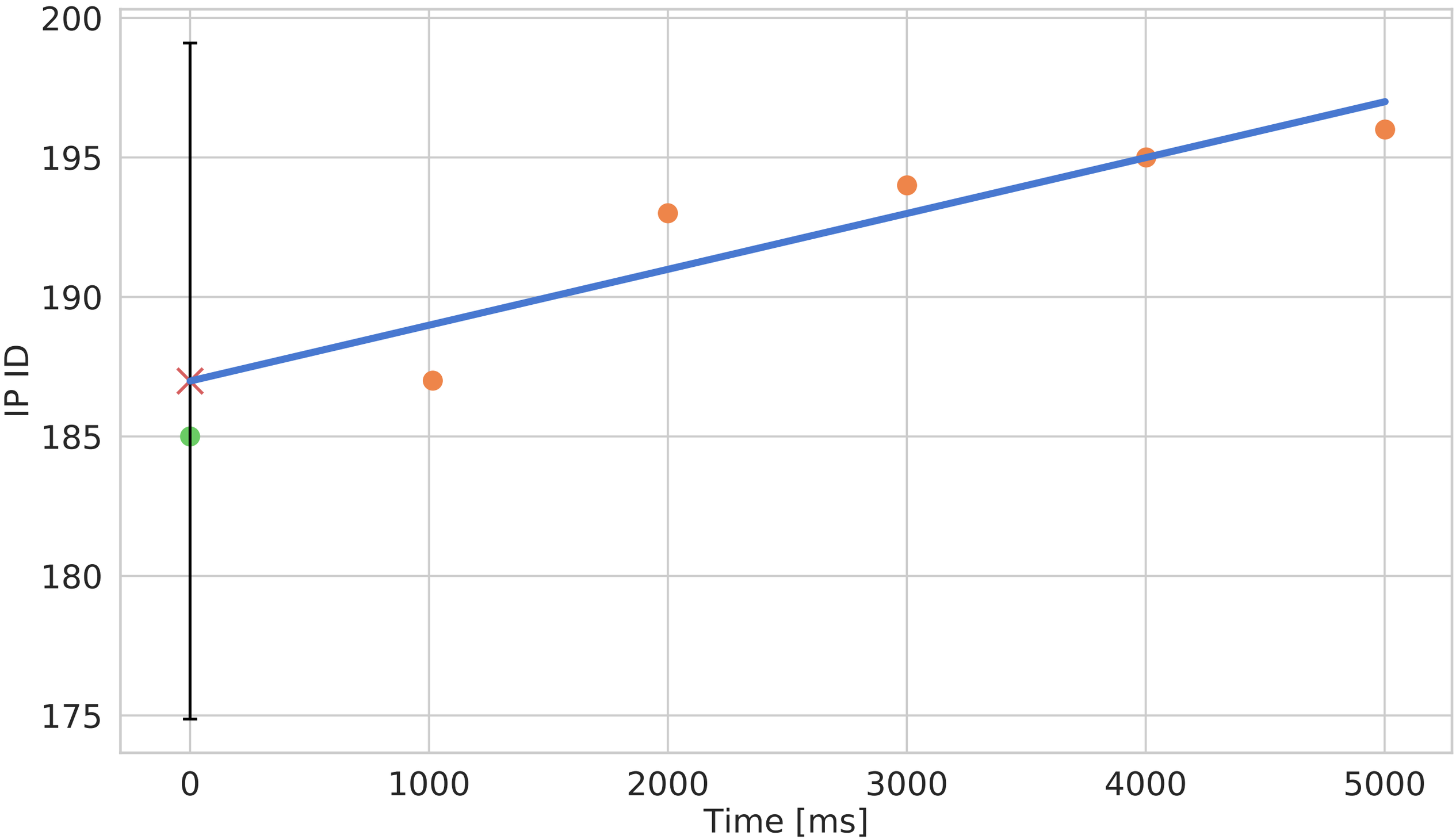
[2] Internet-Scale IPv4 Alias Resolution with MIDAR, *Key et al.*, Transactions on Networking, vol. 21, 2013

# Spoofing-Detection via IP ID Correlation

- *Idea:* Correlate trigger IP ID with the IDs of probe replies
- Identifies valid packets instead of spoofed ones
  - Somewhat inaccurate (e.g., not all hosts reply to probes)
- Previously explored by a CAIDA intern[1]

[1] Design and development of an active probing technique to validate the “source IP address” header field in a live stream of IP packets, *Alessandro Puccetti*, University of Pisa, 2015, *master thesis*

# Example: Consistency Check



# How do we plan to use this?

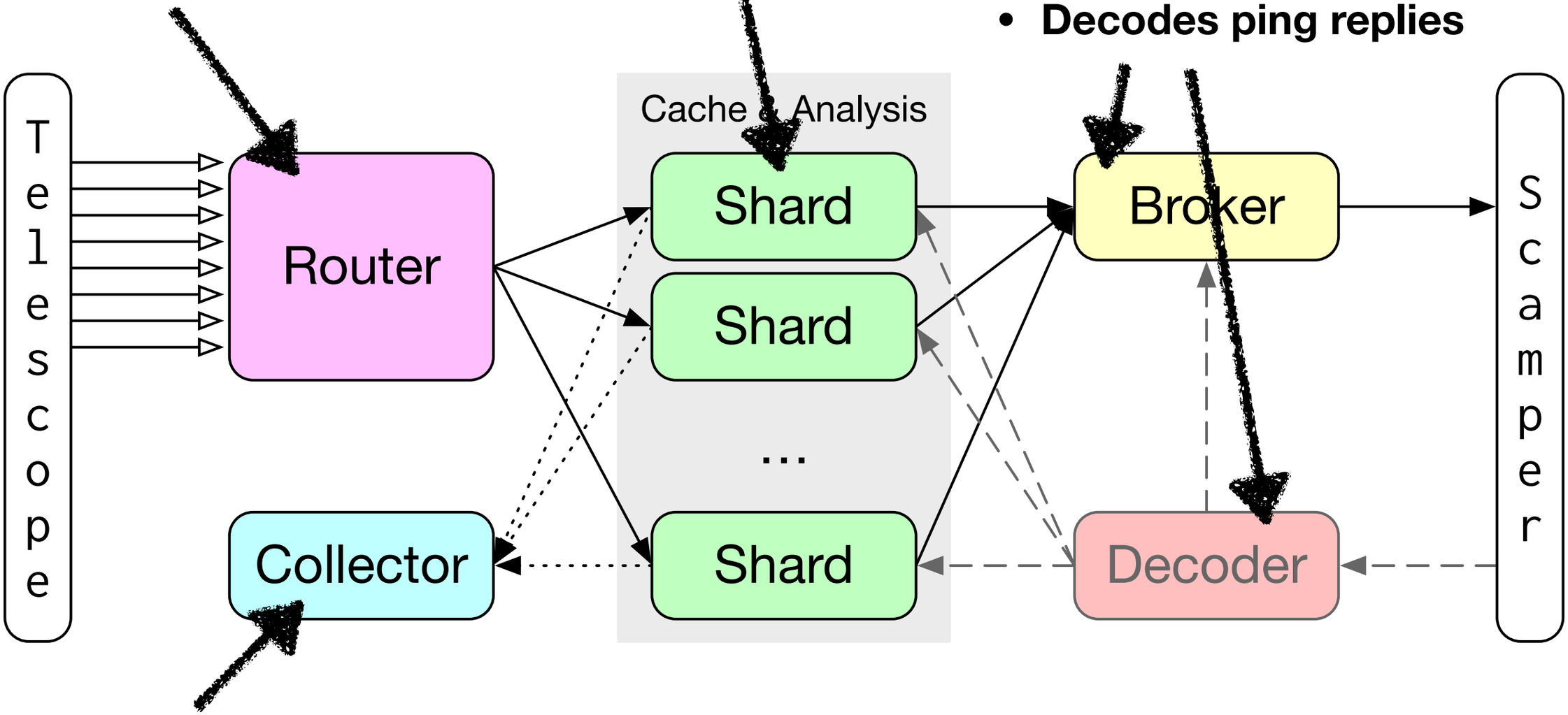
- Build a system that integrates into the telescope backend
- Tag packets to allow filtering during analysis
- Improve the reliability of IBR as resource

# System Overview

- Decides what to probe
- Analyses results
- Caches data reduce workload

- Reads packets in parallel
- Routes same IP to same shard

- Manages scamper instances
- Decodes ping replies




- Collects results
- Writes logs (at the moment)

# Implementation

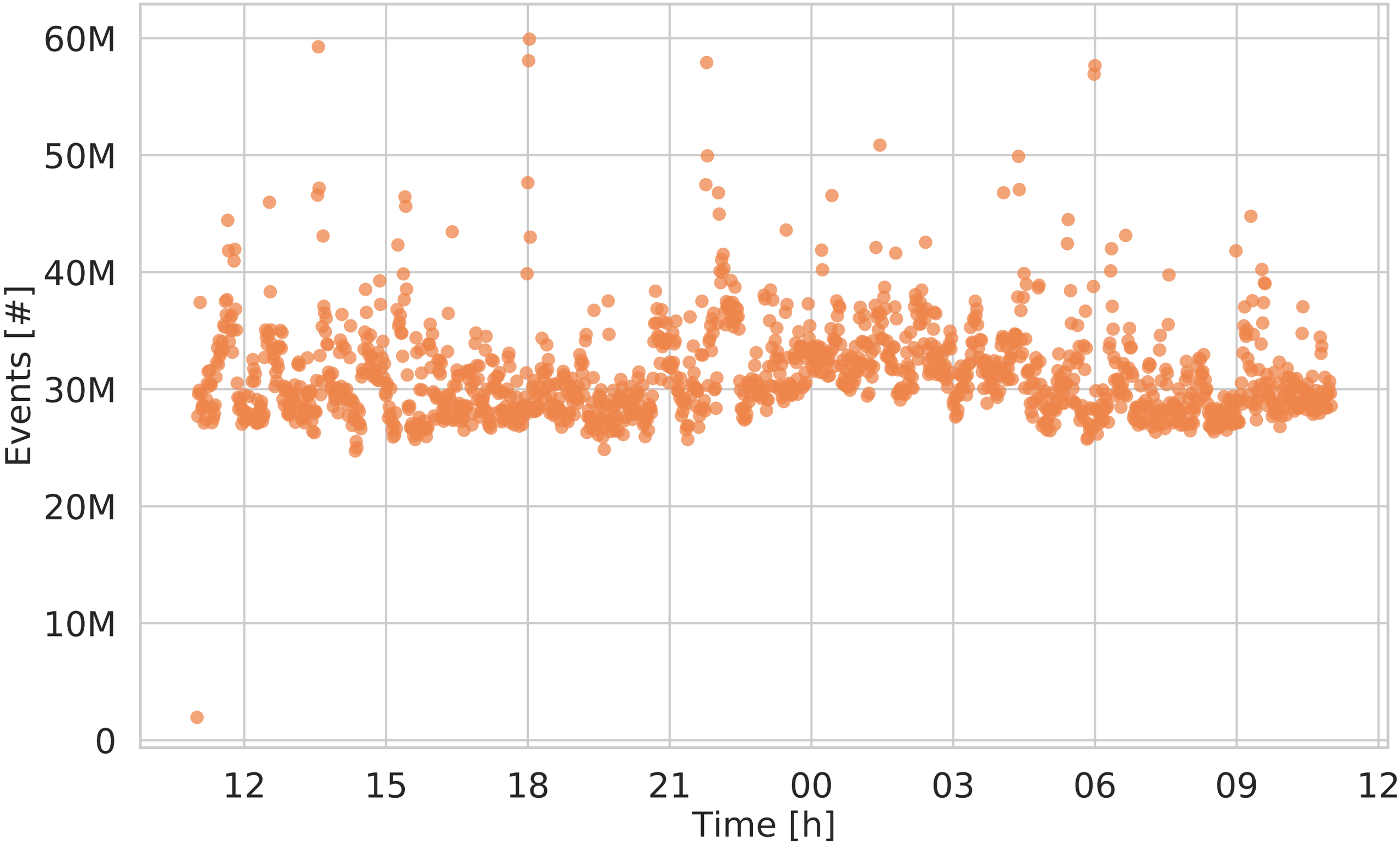
- Implemented in C++11
- Actors as a foundation: C++ Actor Framework[1]
  - Isolated, lightweight entities using message passing
  - Highly scalable runtime environment with a work-stealing scheduler
- Parallel packet ingestion via libtrace[2]
- Probing handled by scamper[3]

[1] Revisiting Actor Programming in C++, *Charousset et al.*, Computer Languages, Systems & Structures 2016, https://github.com/actor-framework/actor-framework/

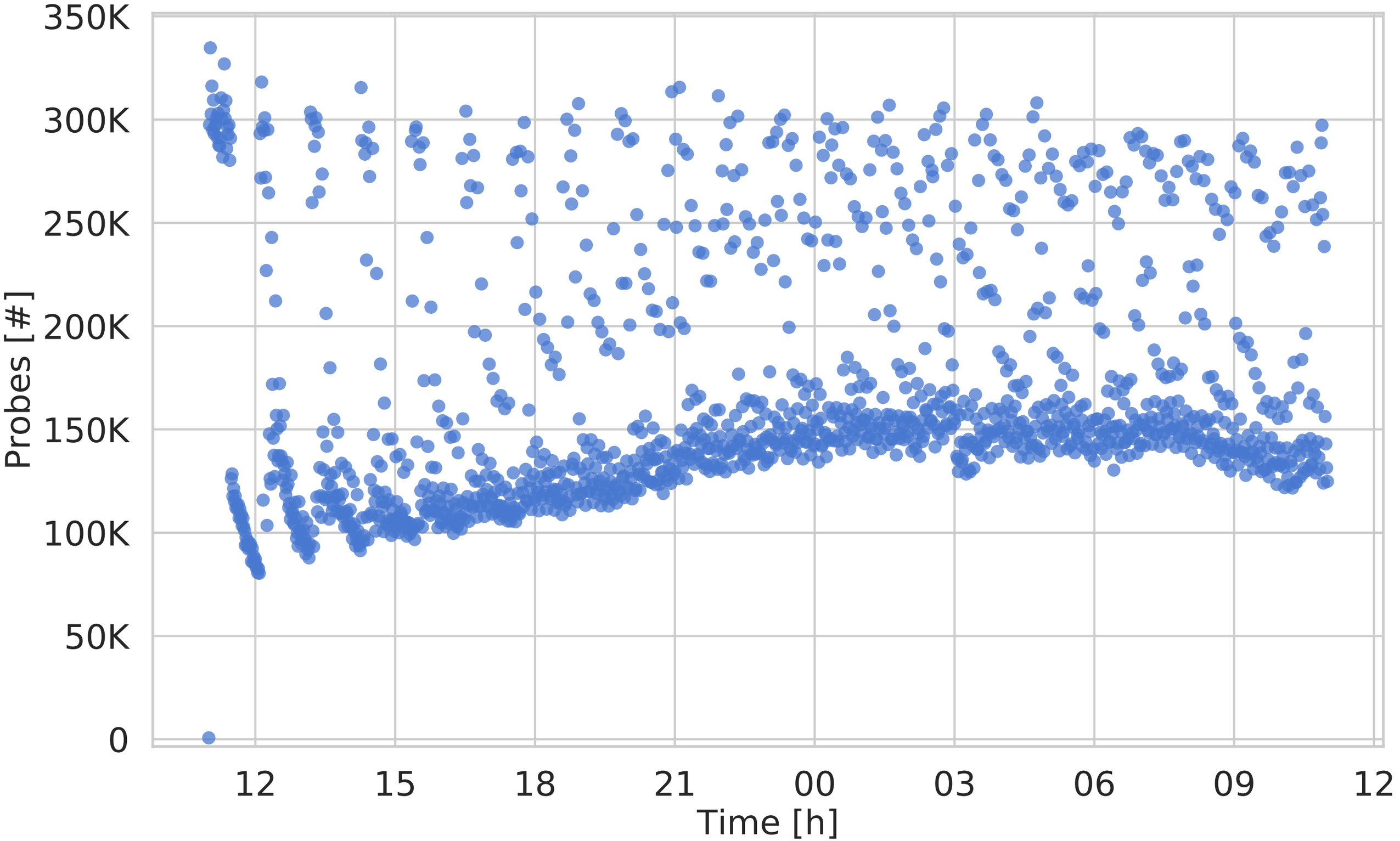
[2] https://github.com/LibtraceTeam/libtrace

[3] Scamper: a Scalable and Extensible Packet Prober for Active Measurement of the Internet, *Matthew Luckie*, IMC'10, https://www.caida.org/tools/measurement/scamper/

# Incoming Events



# Finished Probes

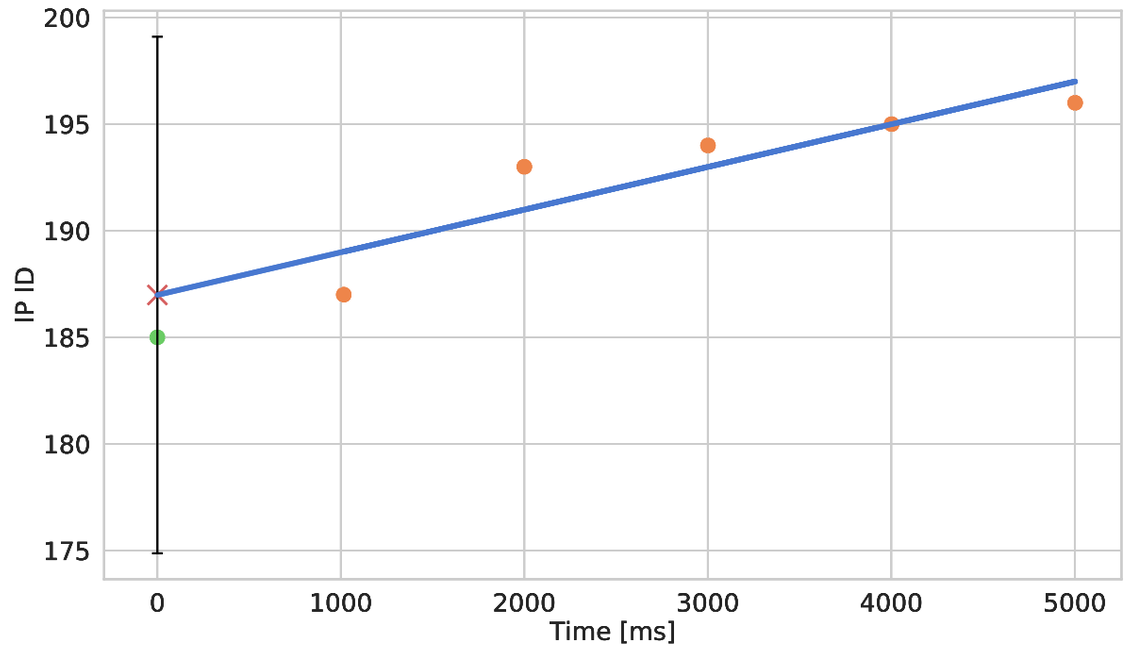


# Analysis

- Send a few probes for each trigger
- Check if probe IP IDs are incrementing monotonically
    - Other observations: random, constant, and no replies
- Drop everything outside a threshold (currently 8000)
- Check consistency

# Linear Regression

- Algorithm
  - Calculate the line of best fit
  - Predict the expected trigger IP ID
  - Use the prediction interval as the acceptable error




- Pro: Established method for predictions
- Contra: The error interval increases quickly with delay

# First Results

| | Absolute | Percentage |
|---|---|---|
| **Events** | 2.083.575 | 100,00 % |
| **Unresponsive** | 1.253.242 | 60,15 % |
| **Responsive** | 830.333 | 39,85 % |
| **Monotonic** | 735.691 | 35,31 % |
| **Within threshold** | 107.237 | 5,15 % |
| **Consistent** | 18.419 | 0,88 % |
| **Consistent of threshold** | | 17,18 % |

# Uhm?

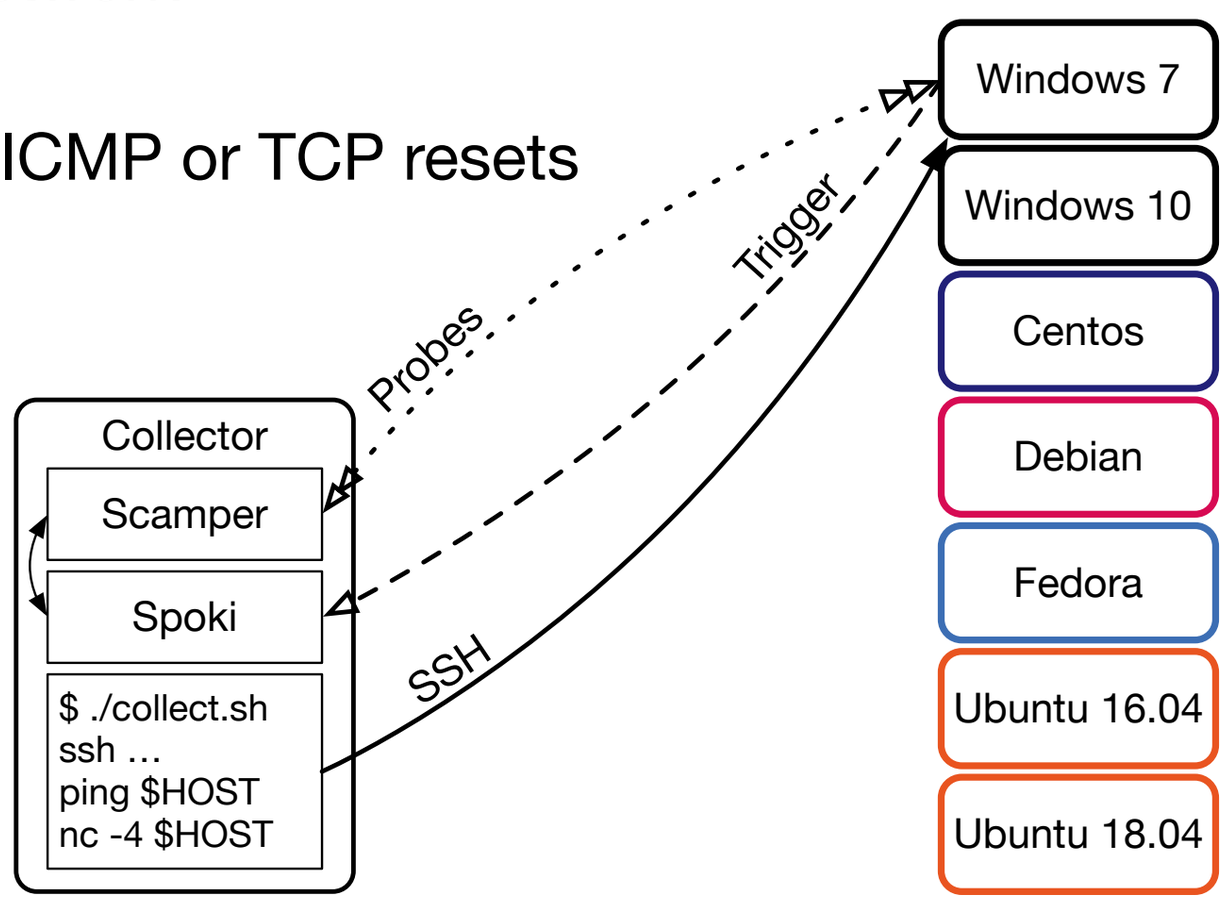
- Found some bugs, but nothing to explain this
- OSes switched to separate counters to improve privacy
  - Linux now has an array of 2048 counters
  - IP addresses and protocol determine which one to use

# The Active Telescope

- Send probes with source address from a few address blocks
- Important: replies must be in the protocol of the trigger
  - ICMP: “easy mode”, send echo requests
  - TCP: “normal mode”
    - Spoof SYN-ACK in response to SYNs
    - Spoof ACK probe with a matching 5-tuple
  - UDP: “hard mode”, replies are service dependent

# Testbed

- Goal
  - A controlled environment to test and validate the idea
  - VMs connected via an internal network
    - Collector does not respond with ICMP or TCP resets
- Scamper on the same host
- Collected 10k probes
- ICMP and TCP work

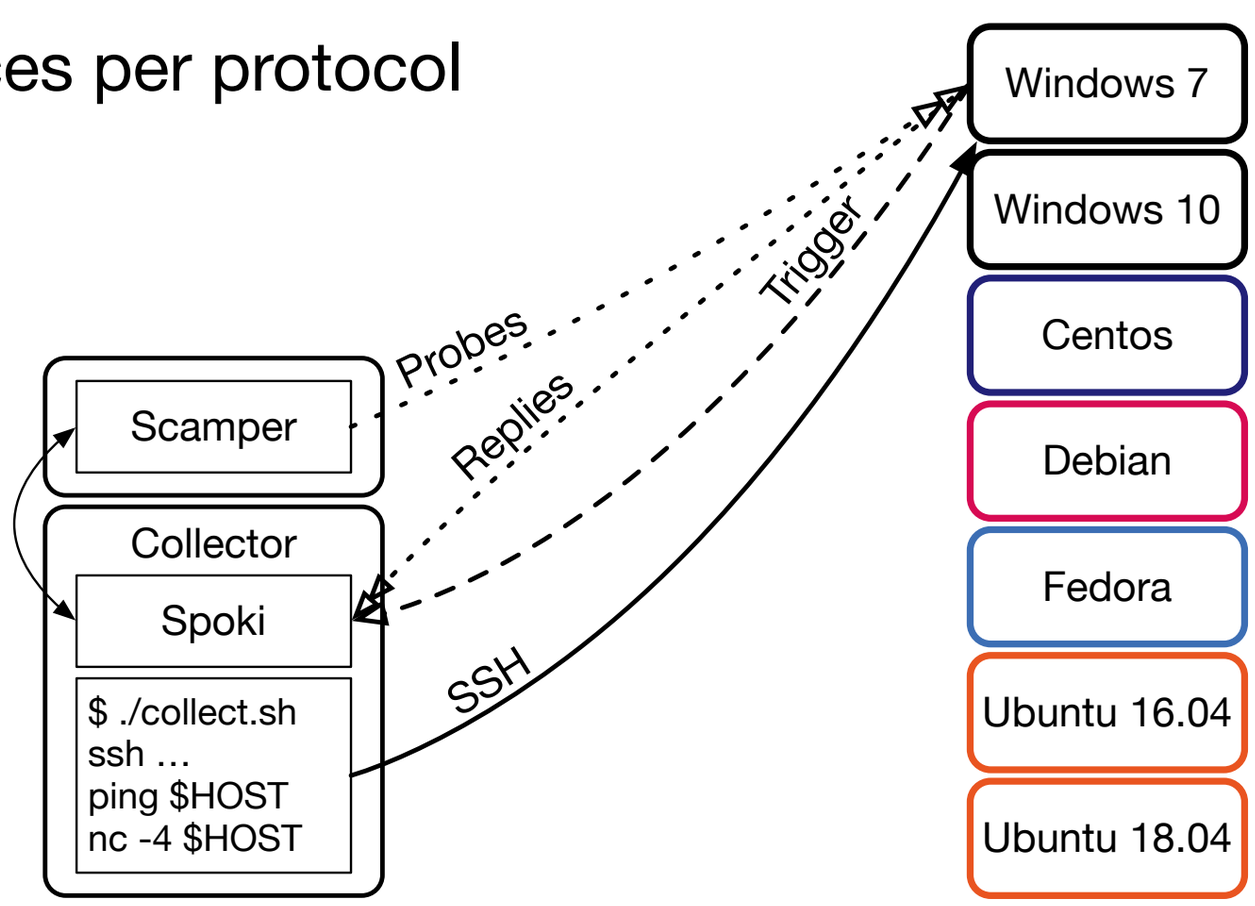


# Recent Work

- Build a testbed with spoofed probes
- Focus on UDP methodology
  - Telescope deployment was delayed
  - UDP is a majority of the traffic

# Testbed with Spoofing

- Changes
  - Move scamper to a separate host
  - Use separate scamper instances per protocol
- Collected 20k probes each
- ICMP validates 97.61%
- TCP validates 100%



# UDP Probing

- UDP is a majority of the traffic
- Responsiveness is (probably) service specific
- There is no connection state we can use
- Closed port returns ICMP “destination unreachable”
- We need UDP responses for the IP ID

# Approaches

- Look how scanners and honeypots handle UDP
- Service-specific probes (e.g., Nmap)
- Send out newlines (e.g., honeytrap)
- Reflect the payload (if it was sent to us it should be valid)

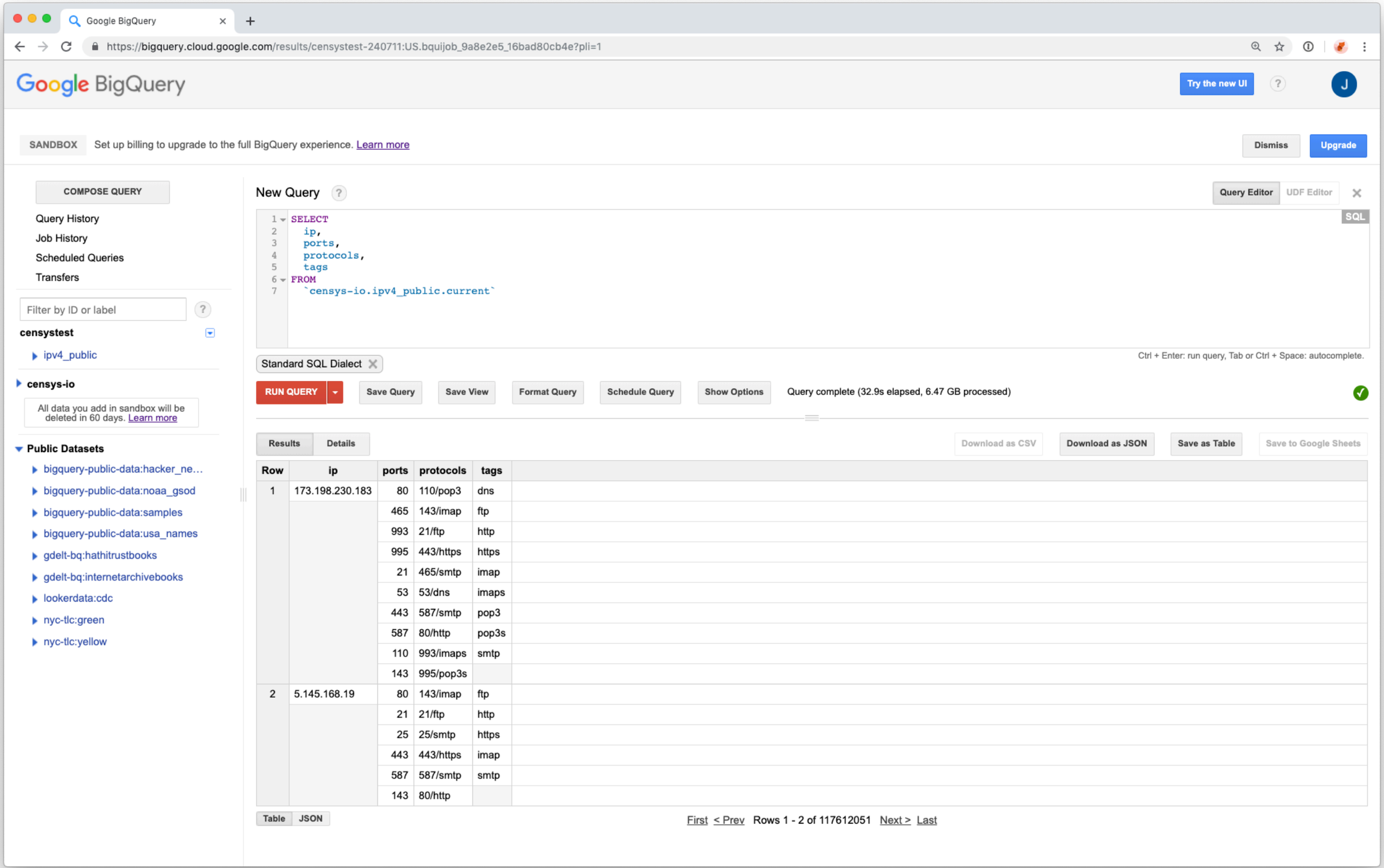
# Port Scanning

- Send generic UDP probe (be aware of ICMP rate limiting)
  - *No replies*: UDP traffic blocked by firewall, NAT, etc.
  - *ICMP reply*:
    - Not everything blocked
    - Ports that don’t provoke a reply are either open or blocked
- Follow up with service-specific probes (such as a DNSStatusRequest)
  - Replies tell you the port is open and runs the expected service
  - Receiving no reply does not give additional information

# Test Data

- Challenge: Find a dataset with targets to probe
- censys.io: “Scanning as a service”
  - Regularly scan about 40 ports
  - Originally a research project and offers researchers free access*
- Self-hosted services
  - Deploy a few services in docker and scan them

# Censys


Google BigQuery
https://bigquery.cloud.google.com/results/censystest-240711:US.bquijob_9a8e2e5_16bad80cb4e?pli=1
Google BigQuery
Try the new UI
J
SANDBOX
Set up billing to upgrade to the full BigQuery experience. Learn more
Dismiss
Upgrade
COMPOSE QUERY
Query History
Job History
Scheduled Queries
Transfers
Filter by ID or label
censystest
ipv4_public
censys-io
All data you add in sandbox will be deleted in 60 days. Learn more
Public Datasets
bigquery-public-data:hacker_ne…
bigquery-public-data:noaa_gsod
bigquery-public-data:samples
bigquery-public-data:usa_names
gdelt-bq:hathitrustbooks
gdelt-bq:internetarchivebooks
lookerdata:cdc
nyc-tlc:green
nyc-tlc:yellow
New Query
Query Editor
UDF Editor
SQL
1 SELECT
2 ip,
3 ports,
4 protocols,
5 tags
6 FROM
7 `censys-io.ipv4_public.current`
Ctrl + Enter: run query, Tab or Ctrl + Space: autocomplete.
Standard SQL Dialect
RUN QUERY
Save Query
Save View
Format Query
Schedule Query
Show Options
Query complete (32.9s elapsed, 6.47 GB processed)
Results
Details
Download as CSV
Download as JSON
Save as Table
Save to Google Sheets
Row ip ports protocols tags
1 173.198.230.183 80 110/pop3 dns
465 143/imap ftp
993 21/ftp http
995 443/https https
21 465/smtp imap
53 53/dns imaps
443 587/smtp pop3
587 80/http pop3s
110 993/imaps smtp
143 995/pop3s
2 5.145.168.19 80 143/imap ftp
21 21/ftp http
25 25/smtp https
443 443/https imap
587 587/smtp smtp
143 80/http
Table
JSON
First < Prev Rows 1 - 2 of 117612051 Next > Last

# Self-Hosted Services

- Use Nmap services as a foundation
- Examined:
  - *Running*: DNS, NTP, SNMP, SLP, DTLS, NFS, ARD, CoAP, memcached
  - *Not running*: SunRPC, NetBIOS, XDMCP, CLDAP, IKE, RIP, IPMI, OpenVPN, Citrix, Radius, Freelancer Game Server, Service Tag Discovery, NAT-PMP, DNS Service Discovery
- Service-specific probes work “well” (small sample size)

# “Insider Knowledge”

- CAIDA receives a lot of
  - DNS responses
  - BitTorrent traffic
- Find a way to handle both (port range + payload analysis)

# How do we plan to use this?

- Real-time detection of large-scale spoofing phenomena
- Validate heuristics and rules already in use
- Check for baseline in our classified traffic
- Monitor baseline changes to identify interesting events

# Next Steps

- Improve our system
  - How to extend the inferences to the entire /8?
  - Find more ideas for UDP
- Work on methodology
  - Compare with other methods of spoofing detection
  - Quantify reliability/expected outcome of different methods
- Can we transfer technique into other contexts?

# Telescope Deployment

- We have a  /24 block at BCIX
  - Continue UDP research
- We (finally) have a /24 block at CAIDA
  - Send RST to close TCP connections we accepted
  - Collect some real-life data for TCP and ICMP

# Research Opportunities

- Examine the impact of “responding” to IBR traffic
  - How does this affect the unsolicited traffic we observe?
  - Does this revert when an address block becomes passive again?
- Accepting TCP connections will provide us with payload
  - Gives additional information, e.g., to attribute packets
  - Data previously available for UDP only