ZMap | Fast Internet-Wide Scanning and its Security Applications

Zakir Durumeric Eric Wustrow J. Alex Halderman

University of Michigan

Internet-Wide Network Studies

Previous research has shown promise of Internet-wide surveys

Mining Ps and Qs: Widespread weak keys in network devices (2012)

EFF SSL Observatory: A glimpse at the CA ecosystem (2010)

Census and Survey of the Visible Internet (2008)

Internet-Wide Network Studies

Previous research has shown promise of Internet-wide surveys

Mining Ps and Qs: Widespread weak keys in network devices (2012)

25 hours acoss 25 Amazon EC2 Instances (625 CPU-hours)

EFF SSL Observatory: A glimpse at the CA ecosystem (2010)

3 months on 3 Linux desktop machines (6500 CPU-hours)

Census and Survey of the Visible Internet (2008)

3 months to complete ICMP census (2200 CPU-hours)


SSL
Observatory

What if…?

What if Internet surveys didn't require heroic effort?

What if we could scan the HTTPS ecosystem every day?

What if we wrote a whole-Internet scanner from scratch?

Introducing ZMap

an open-source tool that can port scan the entire IPv4 address space from just one machine in under 45 minutes with 98% coverage

With Zmap, an Internet-wide TCP SYN scan on port 443 is as easy as:

```
$ zmap -p 443 -o results.txt
34,132,693 listening hosts
(took 44m12s)
```

97% of gigabit Ethernet linespeed

Talk Roadmap

1. **Philosophy and Architecture of ZMap**
2. Characterizing ZMap's Performance
3. Applications of High Speed Scanning
4. Scanning and Good Internet Citizenship

ZMap Architecture

Existing Network Scanners

Reduce state by scanning in batches
- Time lost due to blocking
- Results lost due to timeouts

Track individual hosts and retransmit
- Most hosts will not respond

Avoid flooding through timing
- Time lost waiting

Utilize existing OS network stack
- Not optimized for immense number of connections

ZMap

Eliminate local per-connection state
- Fully asynchronous components
- No blocking except for network

Shotgun Scanning Approach
- Always send *n* probes per host

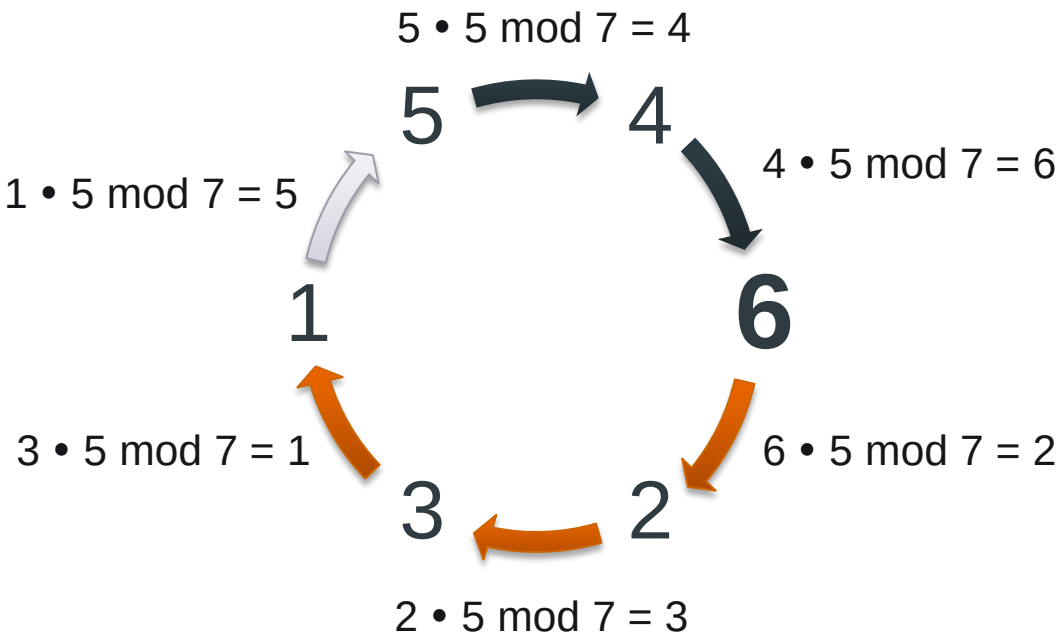
Scan widely dispersed targets
- Send as fast as network allows

Probe-optimized Network Stack
- Bypass inefficiencies by generating Ethernet frames

Addressing Probes

How do we randomly scan addresses without excessive state?

1. Scan hosts according to random permutation
2. Iterate over multiplicative group of integers modulo *p*



Negligible State

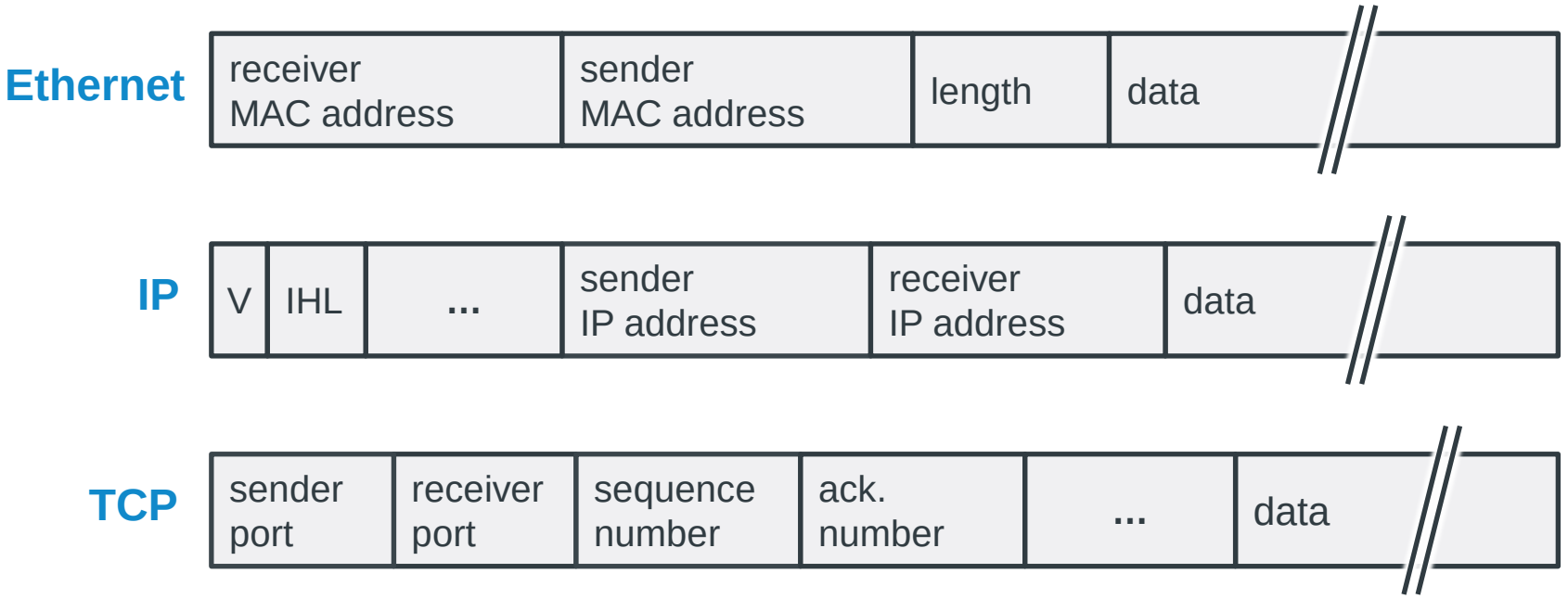
1. Primitive Root
2. Current Location
3. First Address

Validating Responses

How do we validate responses without local per-target state?

Encode secrets into mutable fields of probe packets

that will have recognizable effect on responses

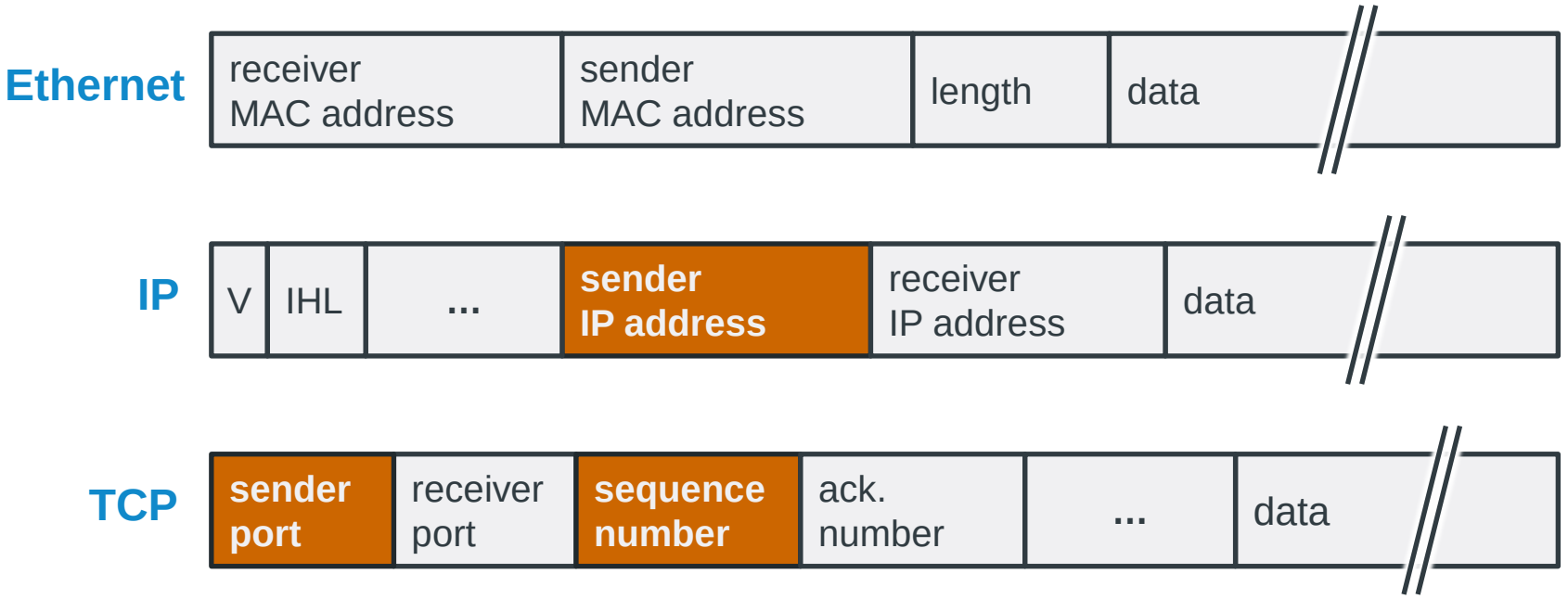


Validating Responses

How do we validate responses without local per-target state?

Encode secrets into mutable fields of probe packets

that will have recognizable effect on responses

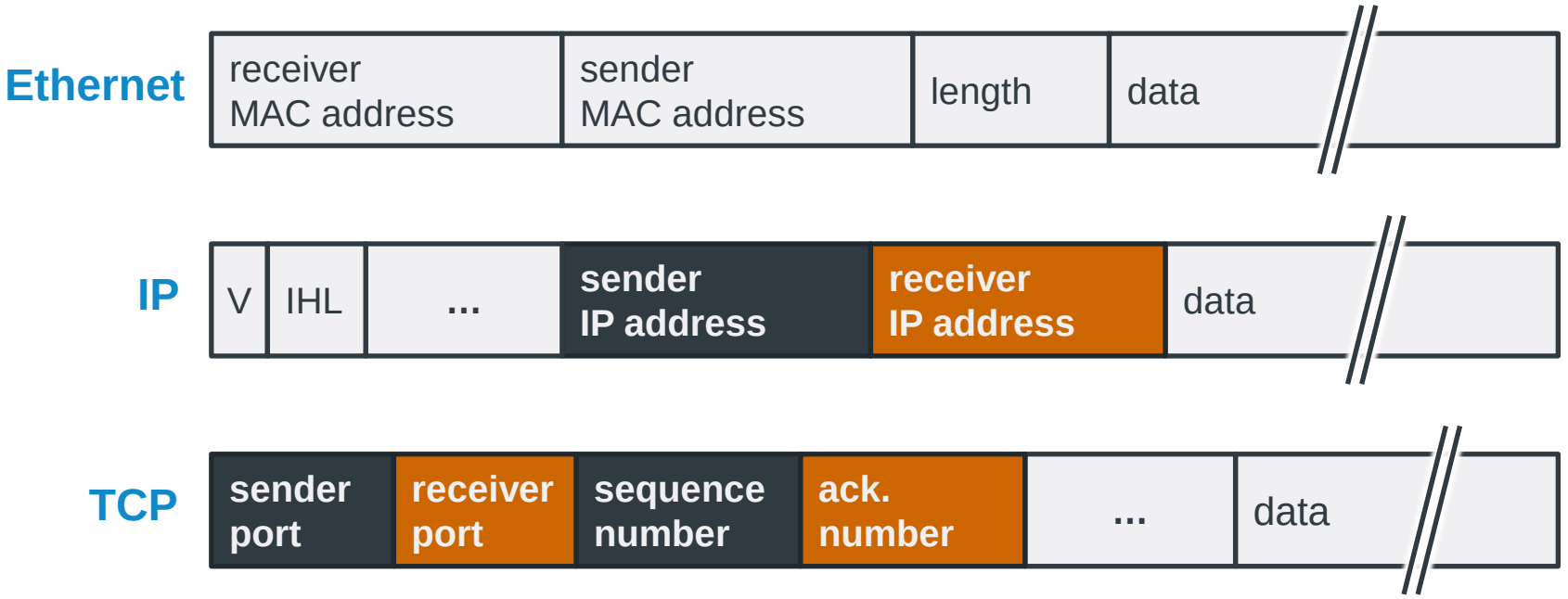


Validating Responses

How do we validate responses without local per-target state?

Encode secrets into mutable fields of probe packets

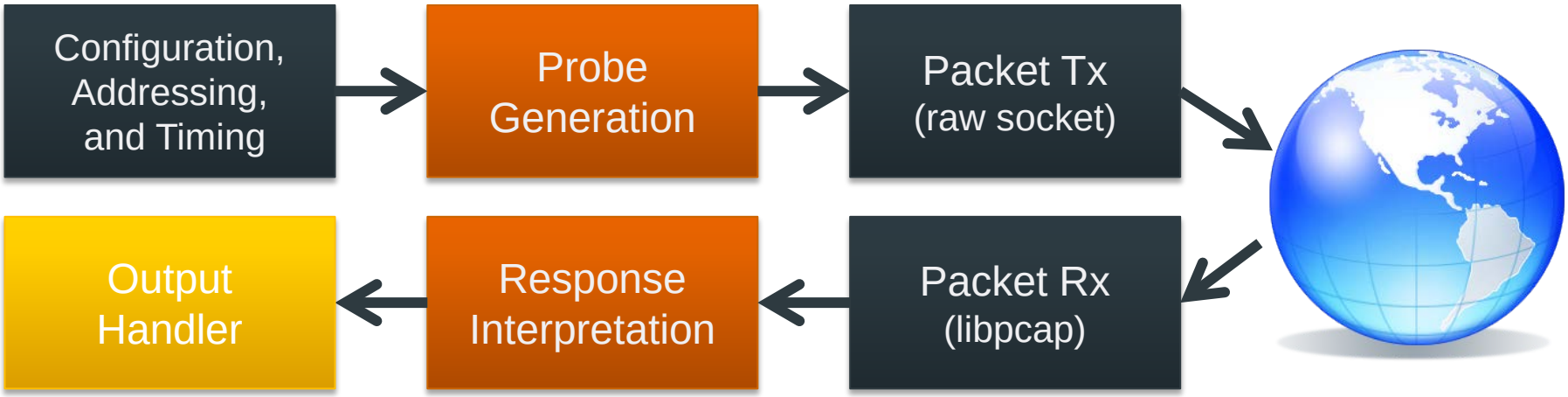
that will have recognizable effect on responses



Packet Transmission and Receipt

How do we make processing probes easy and fast?

1. **ZMap framework** handles the hard work
2. **Probe modules** fill in packet details, interpret responses
3. **Output modules** allow follow-up or further processing



Talk Roadmap

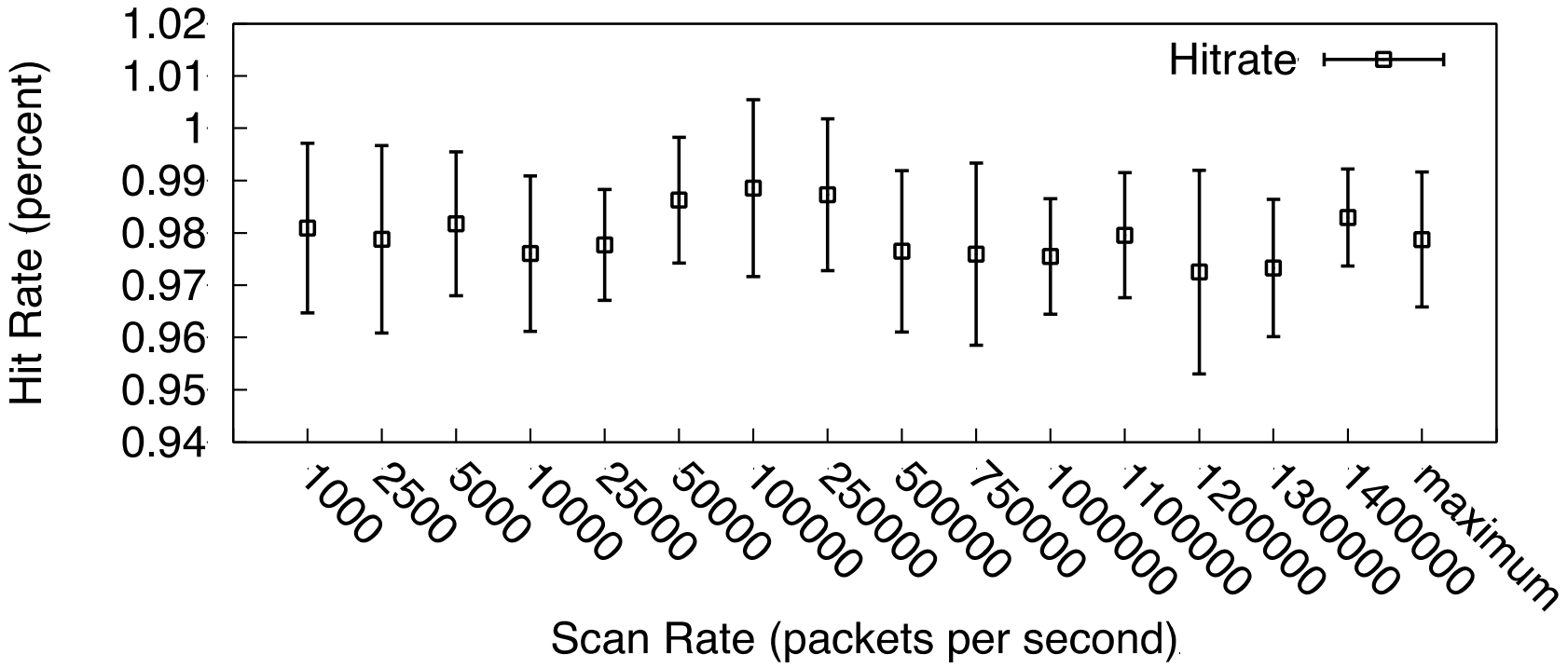
1. Philosophy and Architecture of ZMap
2. **Characterizing ZMap's Performance**
3. Applications of High Speed Scanning
4. Scanning and Good Internet Citizenship

Scan Rate

How fast is too fast?

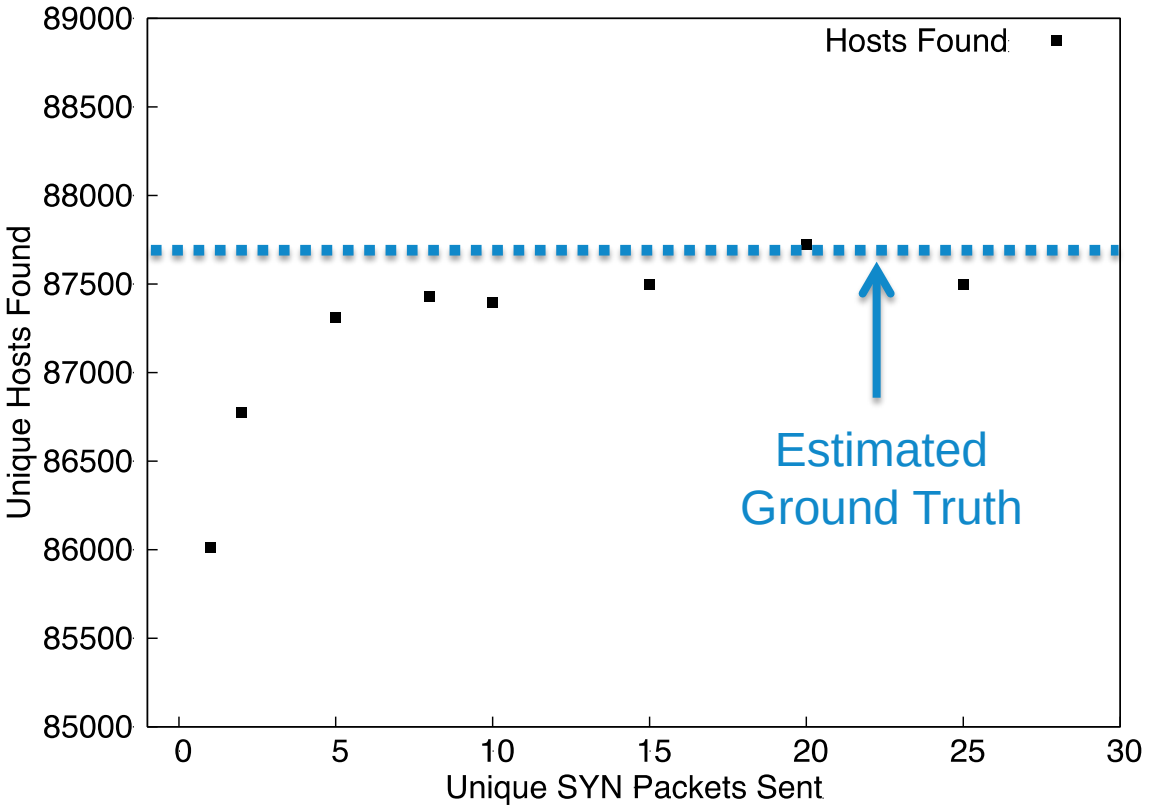
No correlation between hit-rate and scan-rate.

Slower scanning does not reveal additional hosts.



Coverage

Is one probe packet sufficient?



We expect an eventual plateau in responsive hosts, regardless of additional probes.

Scan Coverage

1 Packet:	97.9%
2 Packets:	98.8%
3 Packets:	99.4%

Comparison with Nmap

Averages for scanning 1 million random hosts

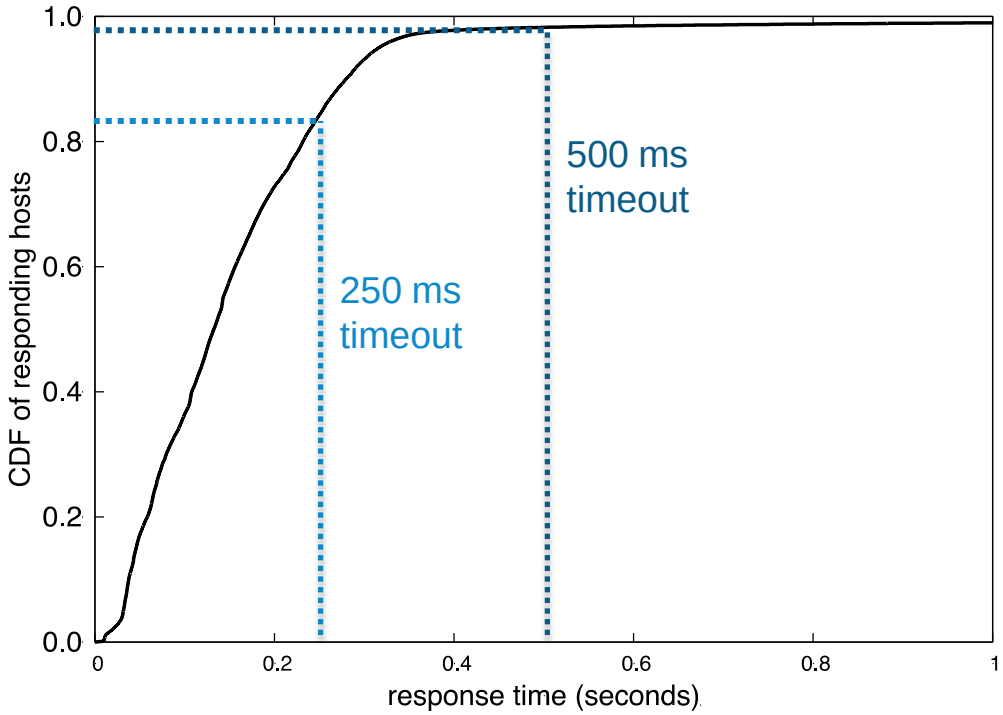
	Normalized Coverage	Duration (mm:ss)	Est. Internet Wide Scan
Nmap (1 probe)	81.4%	24:12	62.5 days
Nmap (2 probes)	97.8%	45:03	116.3 days
ZMap (1 probe)	98.7%	00:10	1:09:35
ZMap (2 probes)	100.0%	00:11	2:12:35

ZMap is capable of scanning more than 1300 times faster than the most aggressive Nmap default configuration (“insane”)

Surprisingly, ZMap also finds more results than Nmap

Probe Response Times

Why does ZMap find more hosts than Nmap?



Response Times

250 ms:	**< 85%**
500 ms:	**98.2%**
1.0 s:	**99.0%**
8.2 s:	**99.9%**

Statelessness leads to both higher performance ***and*** increased coverage.

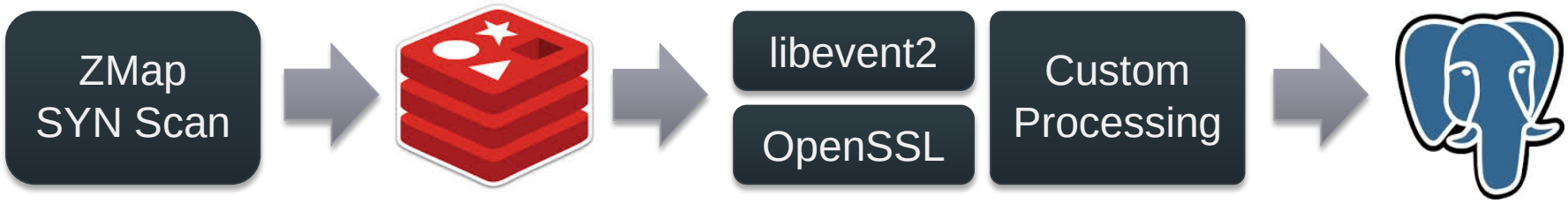
Talk Roadmap

1. Philosophy and Architecture of ZMap
2. Characterizing ZMap's Performance
3. **Applications of High Speed Scanning**
4. Scanning and Good Internet Citizenship

Visibility into Distributed Systems

Gaining near real-time perspective into the CA ecosystem

ZMap enables us to scan the public HTTPS Ecosystem every day

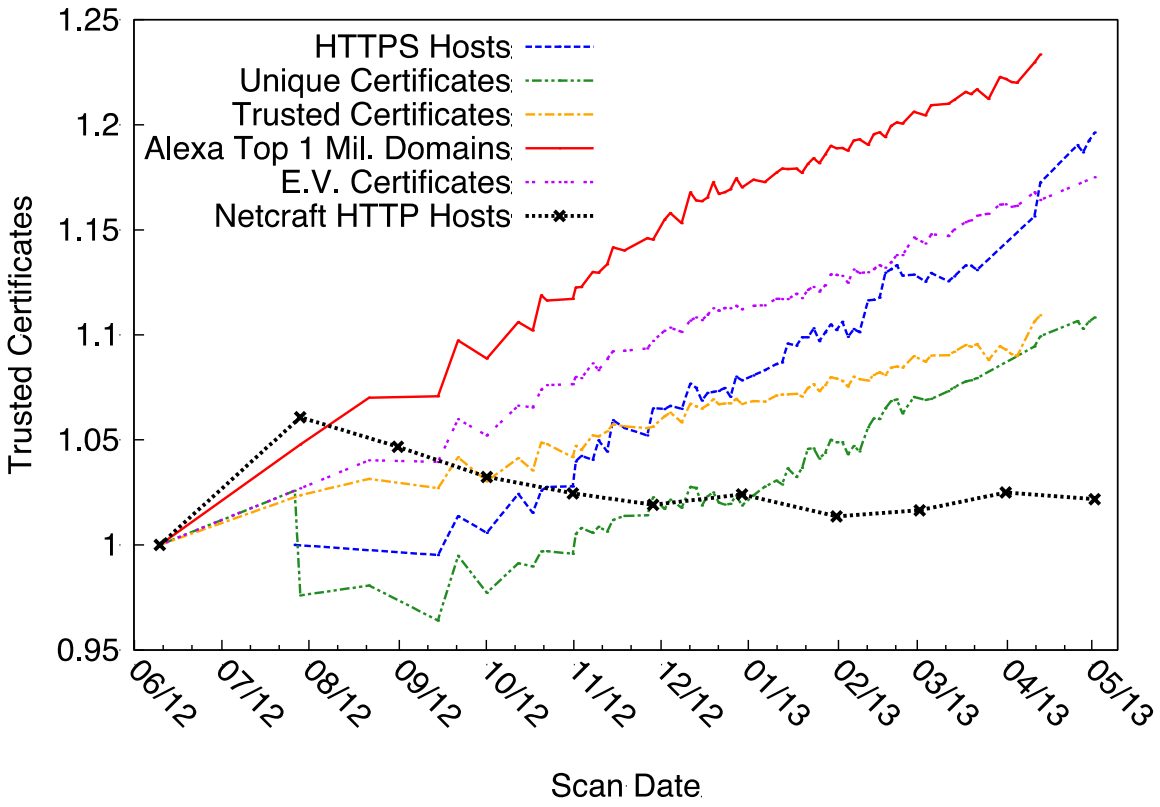


Completed 110 scans of the HTTPS ecosystem in the last year

We collected more than **42 million unique certificates** of which **6.9 million were browser trusted**. Identified 2 sets of misissued CA certificates.

Tracking Protocol Adoption

Examining the growth in global HTTPS adoption



June 2012–May 2013

10% ⇧ HTTPS servers.

23% ⇧ Use on Alexa Top-1M sites.

11% ⇧ Browser-trusted certificates.

Enumerating Vulnerable Hosts

Discovering UPnP Vulnerabilities En Masse

HD Moore disclosed vulnerabilities in several common UPnP frameworks in January 2013.

Under 6 hours to code and run UPnP discovery scan. Custom probe module, 150 SLOC.

We found that 3.34 M of 15.7 M devices were vulnerable.

Compromise possible with a single UDP packet!

Uncovering Hidden Services

Enumerating Unadvertised Tor Bridges

Scanning has potential to uncover unadvertised services

We perform a Tor handshake with public IPv4 addresses on port 9001 and 443

We identified 86% of live allocated bridges with a single scan

Tor has developed *obfsproxy* that listens on random ports to count this type of attack

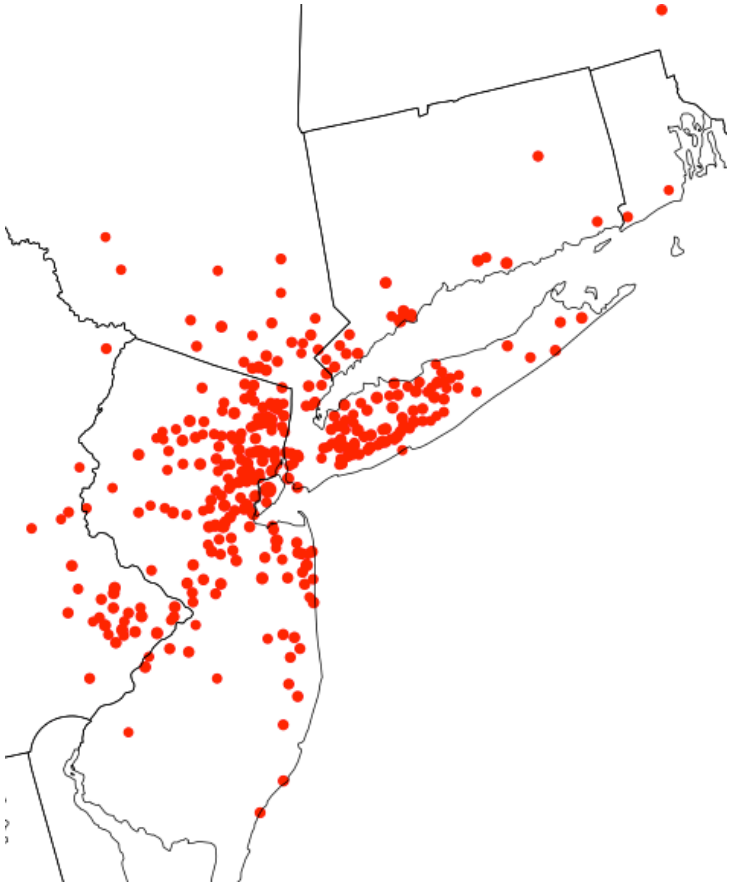
Further ZMap Potential

Further Potential Applications

- Detect Service Disruptions
- Track Adoption of Defenses
- Study Criminal Behavior

Other Security Implications

- Anonymous Communication
- Track users between IP leases



Snapshot of HTTPS outages caused by Hurricane Sandy

Talk Roadmap

1. Philosophy and Architecture of ZMap
2. Characterizing ZMap's Performance
3. Applications of High Speed Scanning
4. **Scanning and Good Internet Citizenship**

Ethics of Active Scanning

Considerations

Impossible to request permission from all owners

No IP-level equivalent to robots exclusion standard

Administrators may believe that they are under attack

Reducing Scan Impact

Scan in random order to avoid overwhelming networks

Signal benign nature over HTTP and w/ DNS hostnames

Honor all requests to be excluded from future scans

User Responses

Over 200 Internet-wide scans over the past year (>1 trillion probes)

Responses from 145 users

Blacklisted 91 entities
(3.7 M total addresses)

15 hostile responses

2 cases of retaliatory traffic

Entity Type	Responses
Small Business	41
Home User	38
Corporation	17
Academic Institution	22
Government	15
ISP	2
Unknown	10
Total	**145**

Future Work

10gigE Network Surveys

TLS Server Name Indication

Scanning Exclusion Standards

IPv6 Scanning Methdology?

Use ZMap to do great research!

Public Release

Releasing ZMap as a fully documented open source project

Downloaded it now from **https://zmap.io**

Scanning the Internet ***really is*** as simple as:

```
$ zmap -p 443 -o results.txt
```

Be sure you have adequate bandwidth and be a good Internet neighbor!

Conclusion

Living in a unique period

IPv4 can be quickly, exhaustively scanned

IPv6 has not yet been widely deployed

ZMap lowers barriers of entry for Internet-wide surveys

Now possible to scan the entire IPv4 address space from **one host** in under **45 minutes** with **98% coverage**

Explored potential applications

Ultimately we hope ZMap enables future research

ZMap | Fast Internet-Wide Scanning and its Security Applications

https://zmap.io

Zakir Durumeric Eric Wustrow J. Alex Halderman

University of Michigan