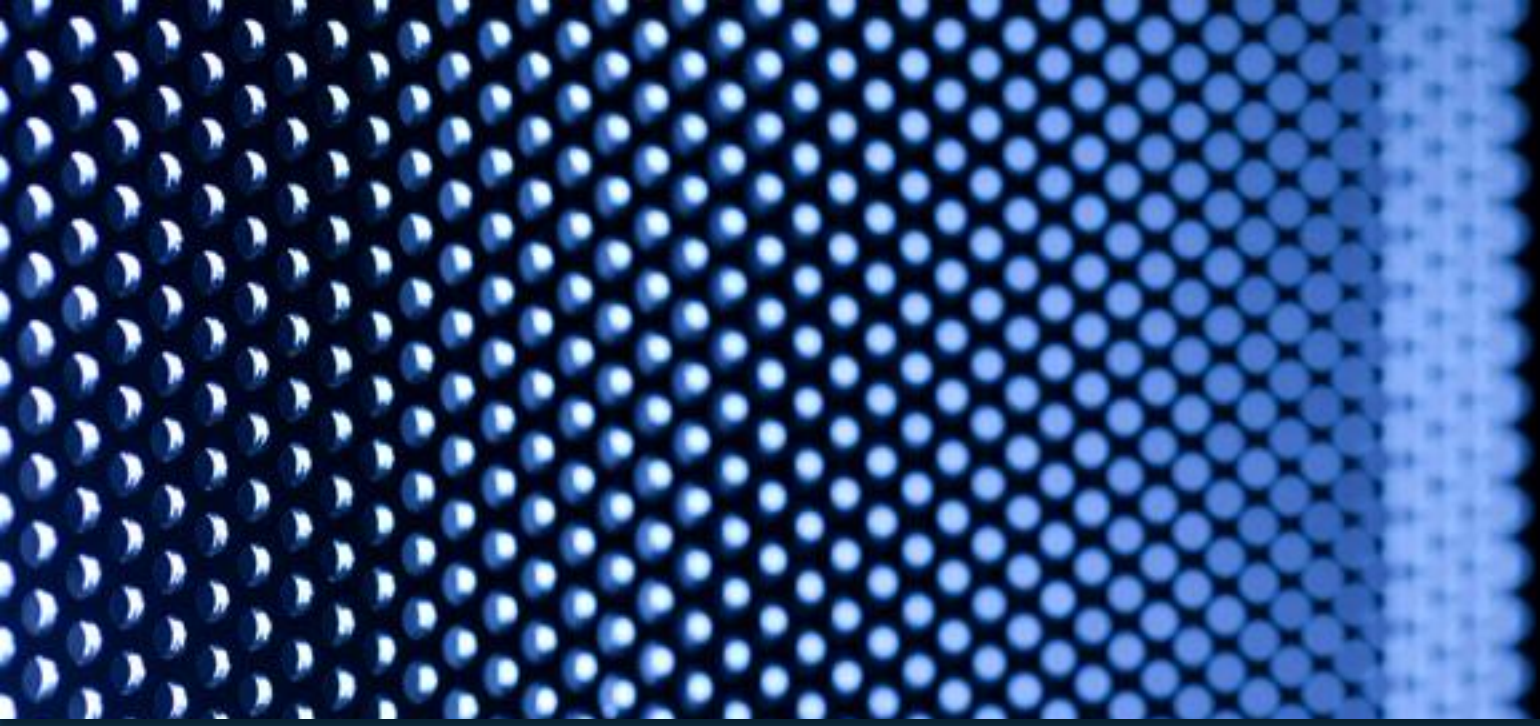


VERISIGN

Detecting and Quantifying Abusive IPv6 SMTP

Casey Deccio

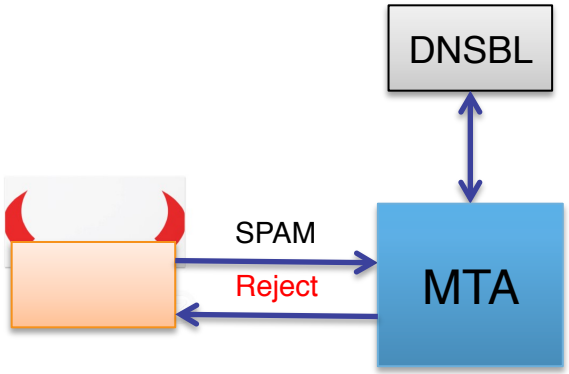
Verisign Labs

NANOG 62, Baltimore, MD

October 6, 2014

Spam, IPv4 Reputation and DNSBL

- Spam is pervasive
  - Annoying (pharmaceuticals)
  - Dangerous (phishing)
- Spam sources are diverse
  - Botnets
  - ISPs with no filtering
- Many IPv4 sources are known and blacklisted
- MTAs subscribe to DNS blacklist
- Reputation-based reject saves computation, reduces risk



Spam, IPv6 and You

- What about IPv6 reputation?
  - Relatively little data
  - Large address space makes traditional blacklist infeasible
- Is there an user risk associated with deploying IPv6-capable MTAs without reputation?
  - Added computation
  - Malicious content allowed to pass
- How can operators quantify risk before deploying IPv6 at their MTAs?

Previous Work

- Steding-Jessen (2009)
  - http://www.cert.br/docs/palestras/certbr-ipv6-national-csirts-meeting2009.pdf
  - Deployed an IPv6 SMTP honeypot using an illegitimate domain (no valid recipients)
  - Little spam found
- Blazquez (RIPE) (2010)
  - https://labs.ripe.net/Members/blazquez/content-spam-over-ipv6
  - IPv6 spam received for production domain was negligible

Spam Honeypot

Email Domain Options

- **Active** (*your domain here!)*
- **Illegitimate** (no legitimate recipients – *ever*)
- **Previously active** (no legitimate recipients – *currently*)

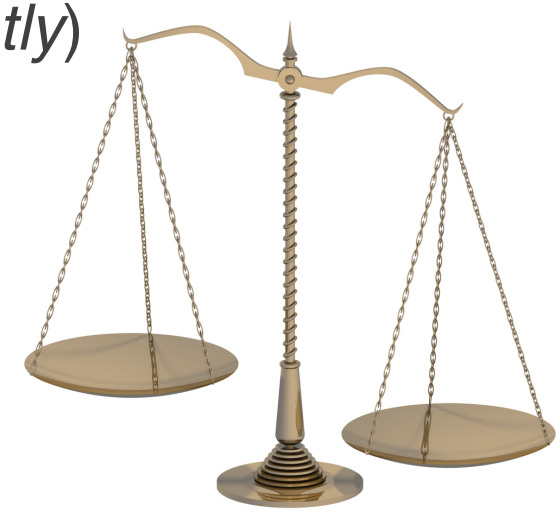


Image credit: Toby Hudson 2011
http://commons.wikimedia.org/wiki/File:Brass_scales_with_cupped_trays.png

Considerations

- Effectiveness
  - Volume of traffic
  - Targeted vs. random
  - Spam/spammer classification
- Security/privacy
  - Circumvention of security filters
  - Disclosure of legitimate emails
- Reliability
  - Impact on production systems

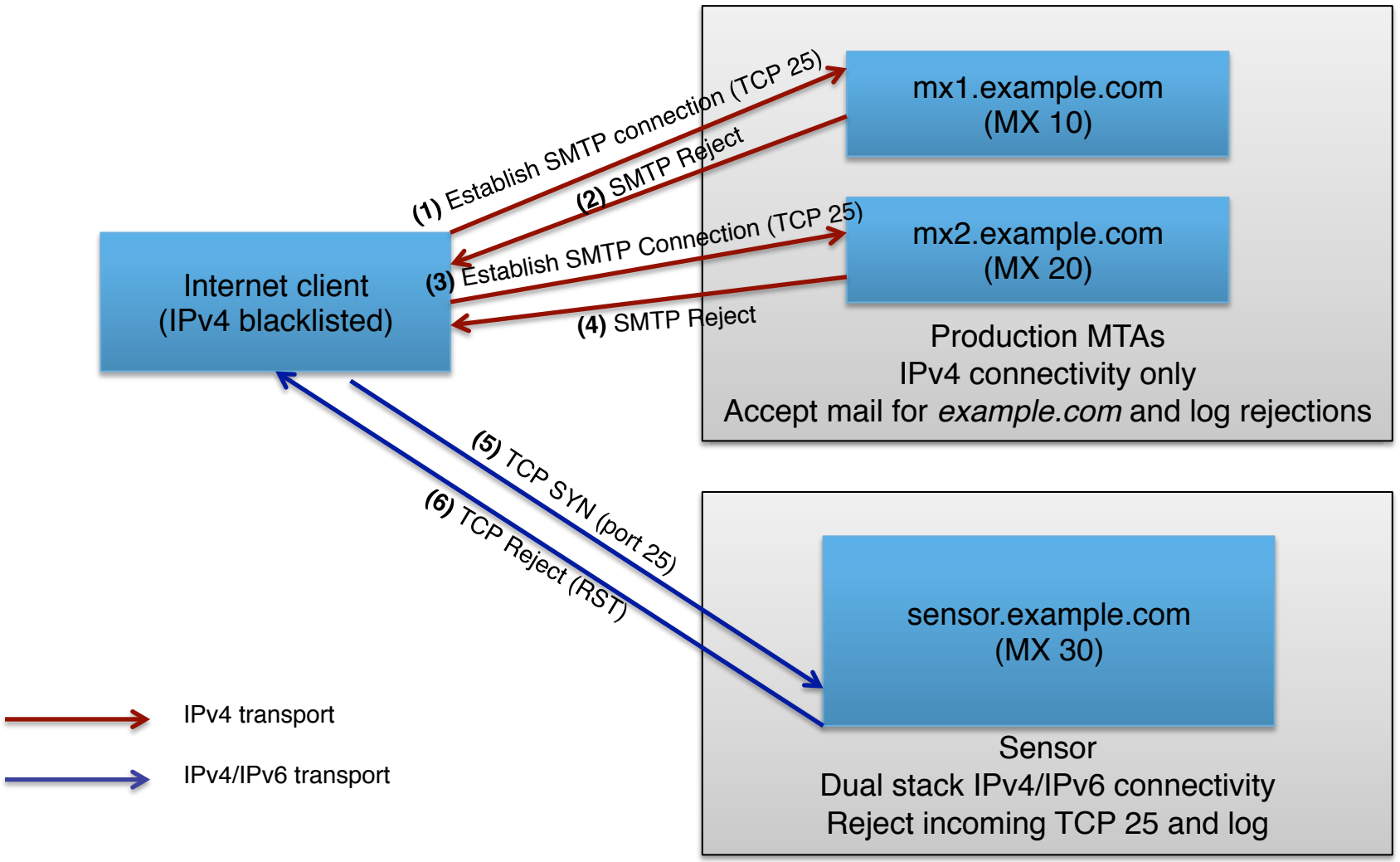
Relevant, Zero-Risk, Abusive IPv6 Measurement

- Active email domain (*your domain here*!)
- Comparatively high traffic resulting from:
  - **Exposure** of domain and email addresses via Web forums, compromised address books, etc.
  - **Value** of legitimate accounts to spammers
- **Relevant value** to users/operators

Do-it-Yourself
Abusive IPv6 SMTP Measurement Instrumentation

- Pre-configuration:
  - Production MTAs are **IPv4 only** and have only A records
  - Production MTAs use DNSBL(s) to **identify** and **reject** IPv4 spam attempts
- Configuration changes:
  1. **Log** IPv4 DNSBL-based rejections at production MTAs
  2. Deploy “sensor MTA” with **both IPv4 and IPv6** and A and AAAA records
  3. **Reject** and log incoming TCP port 25 connection attempts at sensor MTA
  4. Add **higher order MX** to sensor MTA

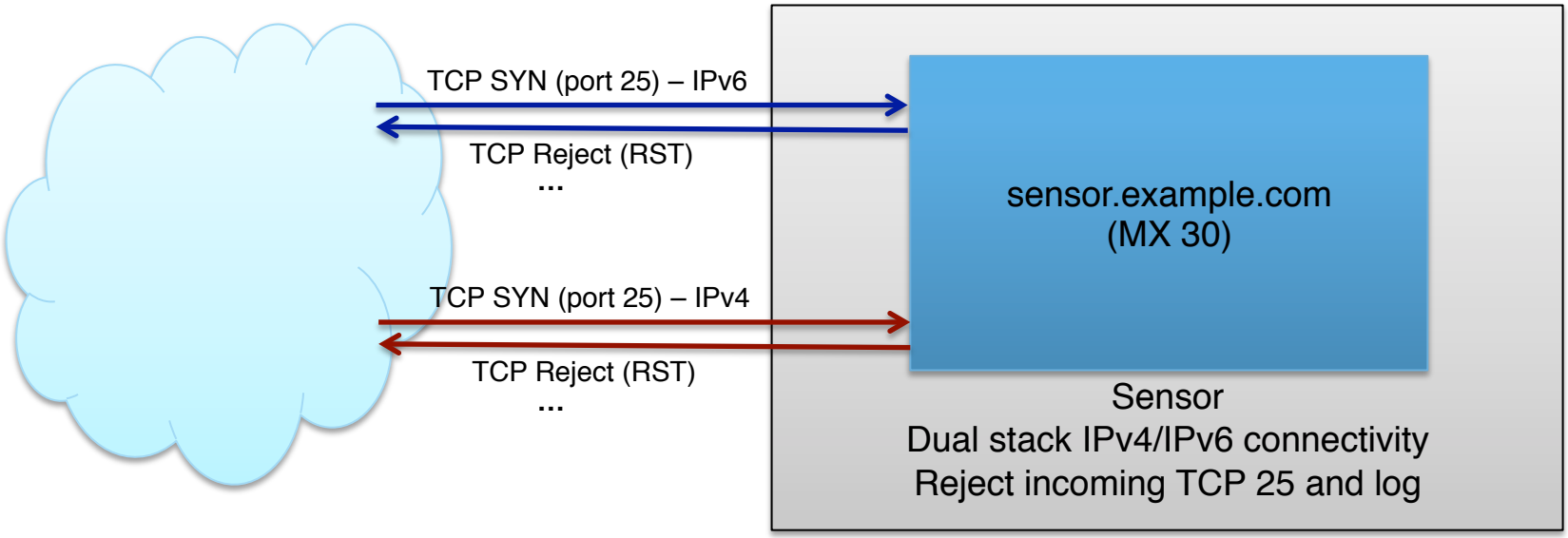
Abusive IPv6 SMTP Measurement Instrumentation – *example.com*



Experimental Architecture Concepts

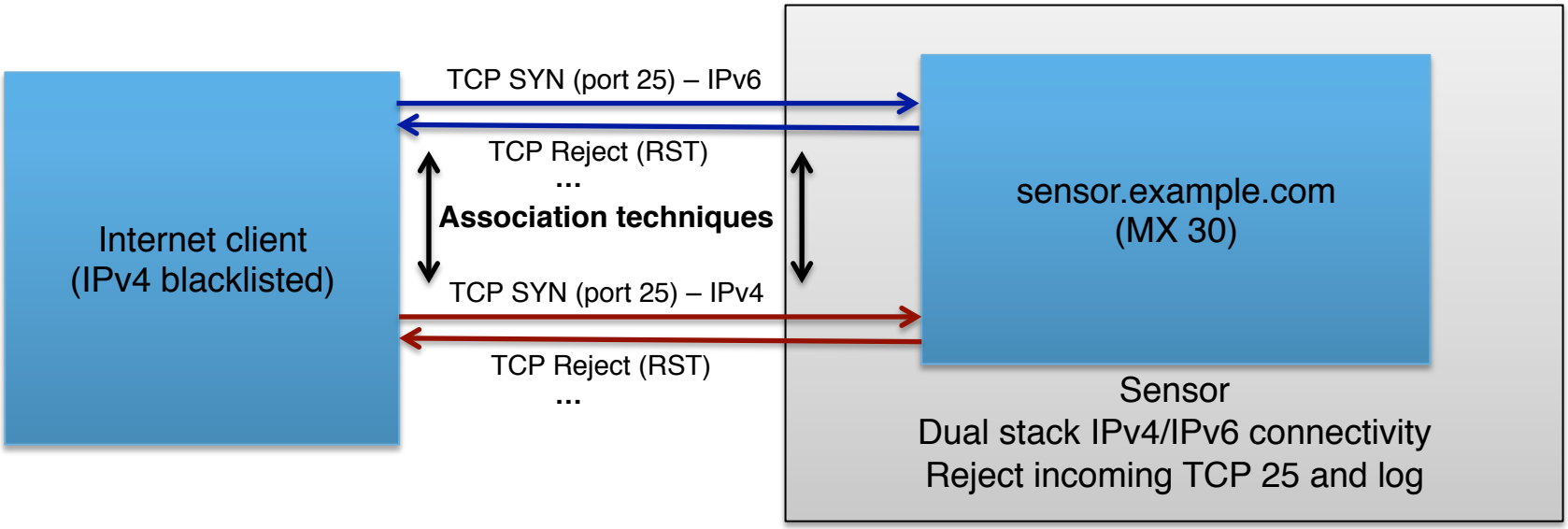
- No mail from known spammers is accepted at the MTAs – over IPv4 or IPv6 **(security)**
- Rejection log at production MTA allows spammers to be identified at sensor MTA **(measurement)**
- No legitimate mail is accidentally delivered to the sensor MTA **(security/stability/privacy)**
- IPv6/IPv4 addresses can be associated for senders willing to attempt delivery both IPv6 and IPv4 **(measurement)**
- Legitimate senders continue to send to production MTAs first **(stability)**

Identifying Spammers at Sensor



- IPv4 spammers are known – due to DNSBL and rejection log at primary MTA
- The challenge is identifying IPv6 spammers

IPv4/IPv6 Address Association at Sensor

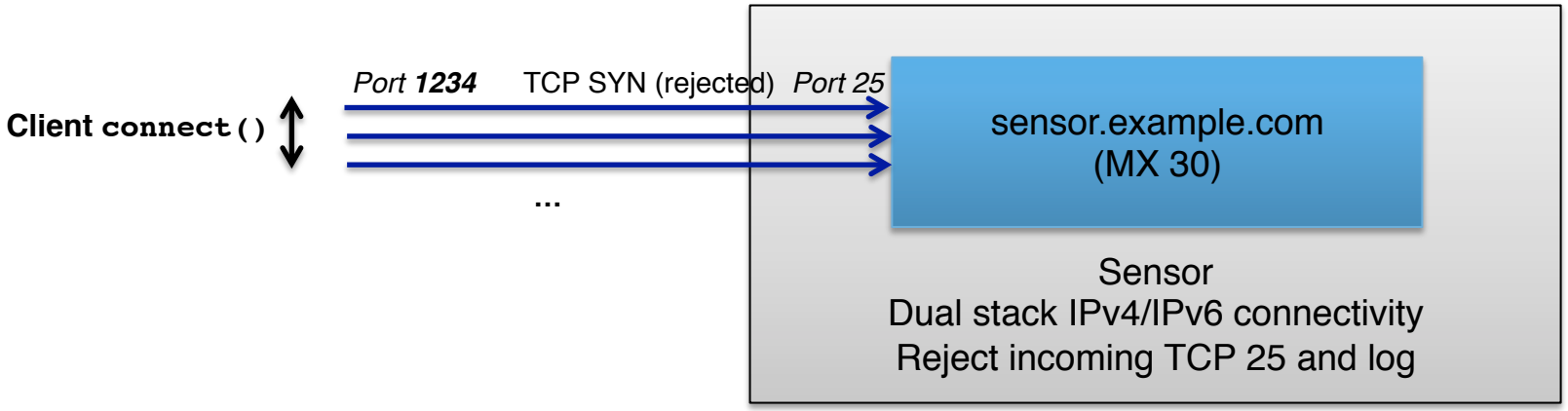


- Identifying IPv6 spammers becomes a game of association with (blacklisted) IPv4 addresses
  1. Associate related SYNs of same `connect()` attempt
  2. Associate `connect()` attempts from same host

Experimental Architecture Caveats

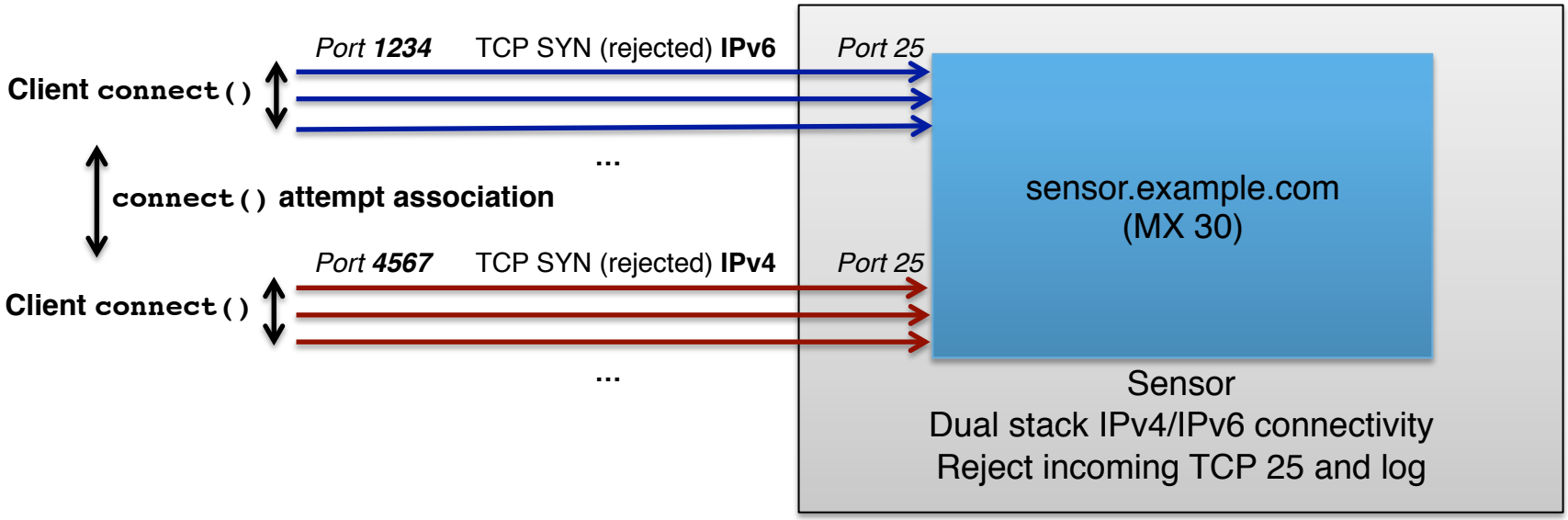
- No message content; spammers identified by association with reject logs
- Spammers don’t necessarily follow prioritized MX ordering
- Spammers don’t necessarily try both IPv4 and IPv6 (i.e., following all addresses in `getaddrinfo()`)
- Network protocols are independent; ground truth is difficult to obtain with only server-side observation

Naïve SYN Association by `connect()`



- Group SYNs by **same source IP/source port** within 25-second sliding window
- Result: "`connect()` attempt"

Naïve `connect()` Attempt Association



- Apparently embedded IPv4 host (last 32 bits – especially with self-addressed 6to4 addresses)
- DNS PTR record
- 6to4 gateway – embedded in 6to4 IPv6 address
- Inferred OS – using `p0f` for TCP fingerprinting
- ASN – from Team Cymru's IP-to-ASN lookup tool

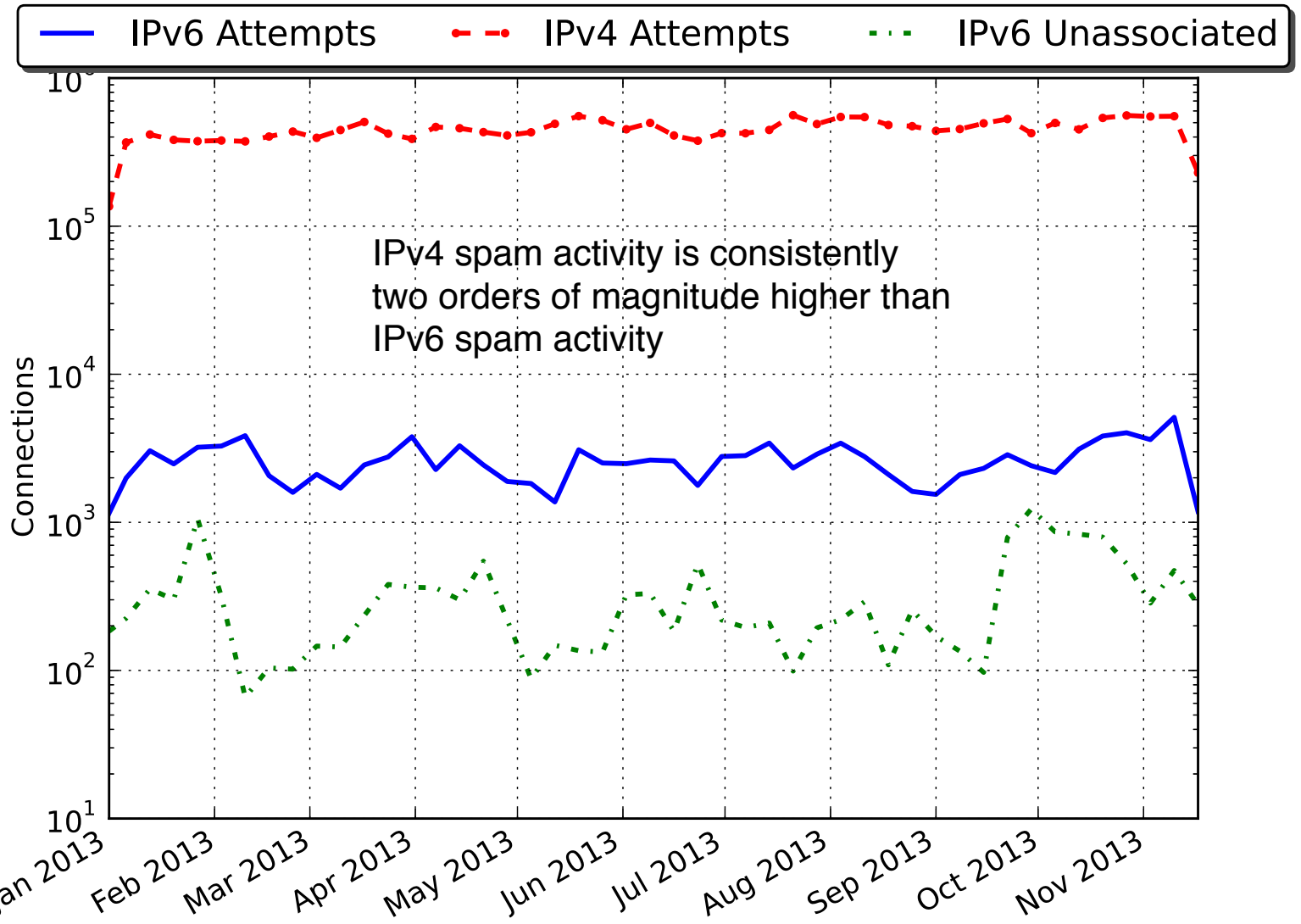
IPv4/IPv6 Preference and `getaddrinfo()`

- RFC 3484 (simplified)
  - If client has **global IPv6 address**, and destination is **global IPv6**
    - Preference ordering: **IPv6, IPv4**
  - If client has only **6to4 IPv6 address** (2002::/16), and destination is **global IPv6**
    - Preference ordering: **IPv6, IPv4**
- RFC 6724 updates (obsoletes RFC 3484)
  - If client has only **6to4 IPv6 address** (2002::/16), and destination is **global IPv6**
    - Preference ordering: **IPv4, IPv6**
- `getaddrinfo()` behavior
  - Windows 7 – conforms to RFC 6724
  - Linux (3.2) – conforms to RFC 6724
  - Mac OS X (10.9) – conforms to RFC 6724

Experimental Architecture – Prototype Results

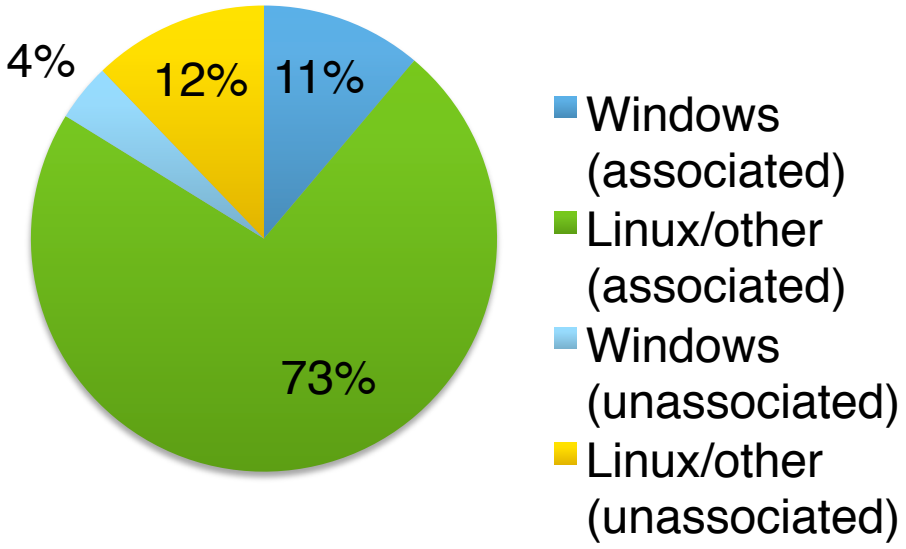
- Production email domain with ~10K users
- Traffic captured Jan – Nov, 2013
- For non-6to4 addresses, IPv6 `connect()` attempts associated with **subsequent** IPv4
- For 6to4 addresses, IPv6 `connect()` attempts associated with **previous** matching IPv4
- OS identification by `p0f`

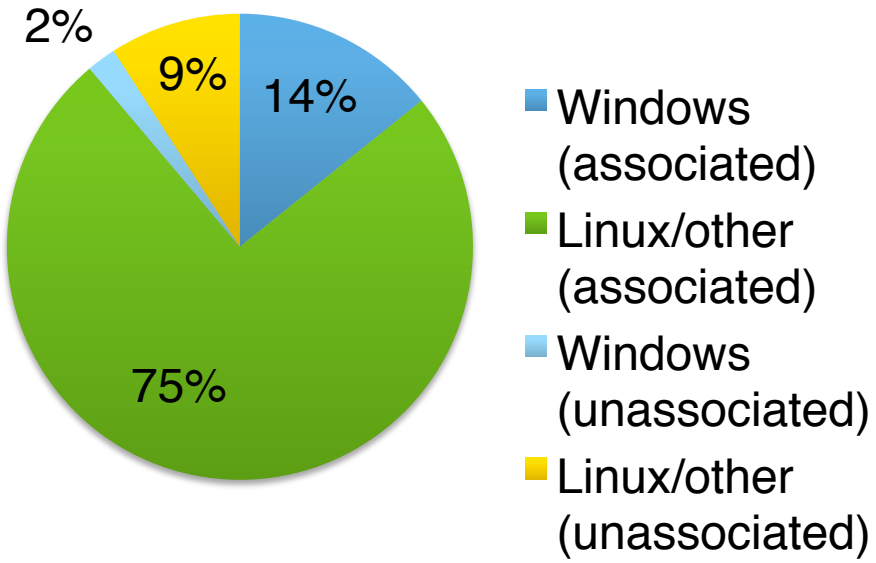
`connect()` Attempts From Spammers Over Time



IPv6 Spammer OSes

	IPv6 Hosts		IPv6 Attempts	
	Associated	**Unassociated**	**Associated**	**Unassociated**
Windows	293 (11%)	105 (4.0%)	18492 (14%)	2652 (2.0%)
Linux	1900 (72%)	317 (12%)	96976 (75%)	11842 (9.1%)
Other	7 (0.27%)	3 (0.11%)	64 (0.05%)	9 (0.00%)



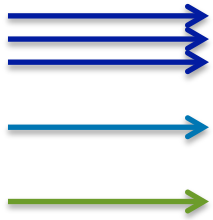


IPv6 Address Types of Spammers

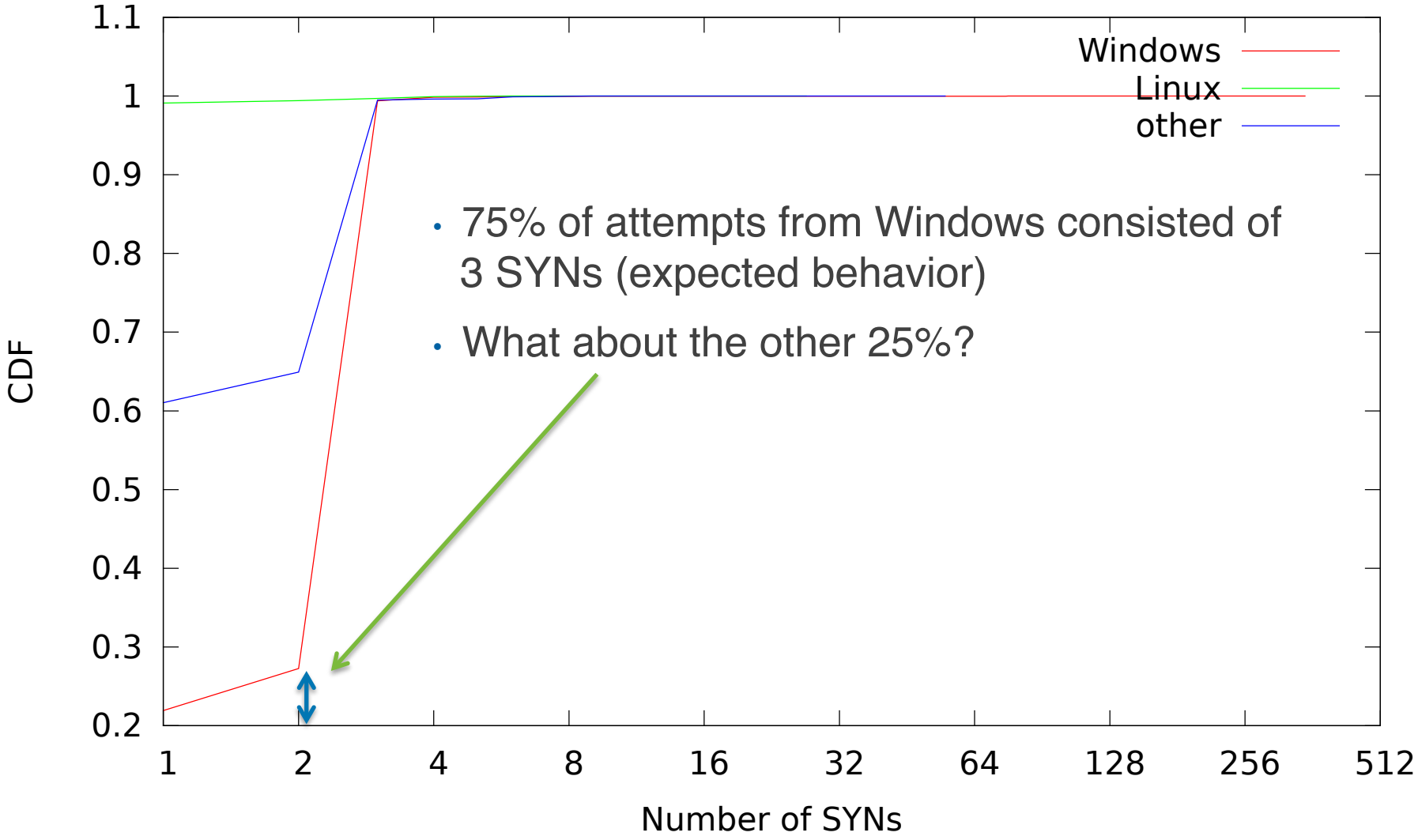
	IPv6 Hosts		IPv6 Attempts	
	Associated	**Unassociated**	**Associated**	**Unassociated**
6to4	252 (7.5%)	63 (1.9%)	16750 (13%)	1408 (1.0%)
Other	2536 (76%)	494 (15%)	101169 (76%)	14135 (11%)
EUI-64	533 (16%)	142 (4.2%)	35888 (27%)	7241 (5.4%)
Embedded IPv4	621 (19%)	108 (3.2%)	36074 (27%)	2387 (1.2%)
Other	1634 (49%)	307 (9.2%)	45967 (34%)	5915 (4.4%)

OS-specific `connect()` Behavior

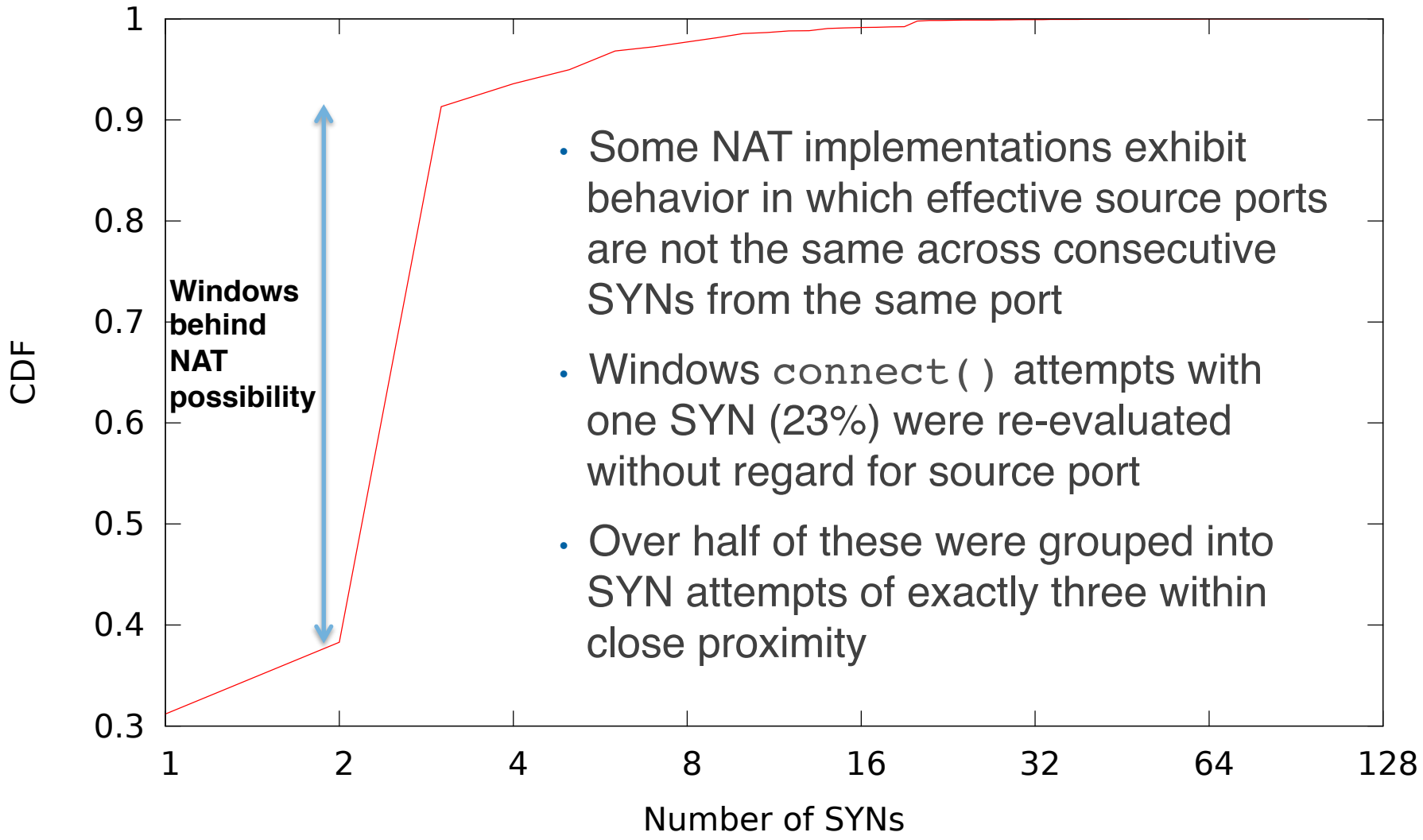
- Different default behaviors across OSs in response to TCP RST
  - Windows XP/7– sends three SYNs – same source port
  - Linux (3.2) – sends single SYN
  - Mac OS X (10.9) – sends single SYN



Number of SYNs for Each Inferred `connect()` Attempt



Windows behind NAT?



- Some NAT implementations exhibit behavior in which effective source ports are not the same across consecutive SYNs from the same port
- Windows `connect()` attempts with one SYN (23%) were re-evaluated without regard for source port
- Over half of these were grouped into SYN attempts of exactly three within close proximity

OS-specific IPv4/IPv6 TCP Source Port Allocation

- For close proximity requests, ephemeral TCP source ports are often allocated sequentially
  - Windows XP/7 – IPv4/IPv6 share the same ephemeral port pool
  - Linux (3.2) – IPv4/IPv6 use distinct ephemeral port pools
  - Mac OS X (10.9) – IPv4/IPv6 use distinct ephemeral port pools

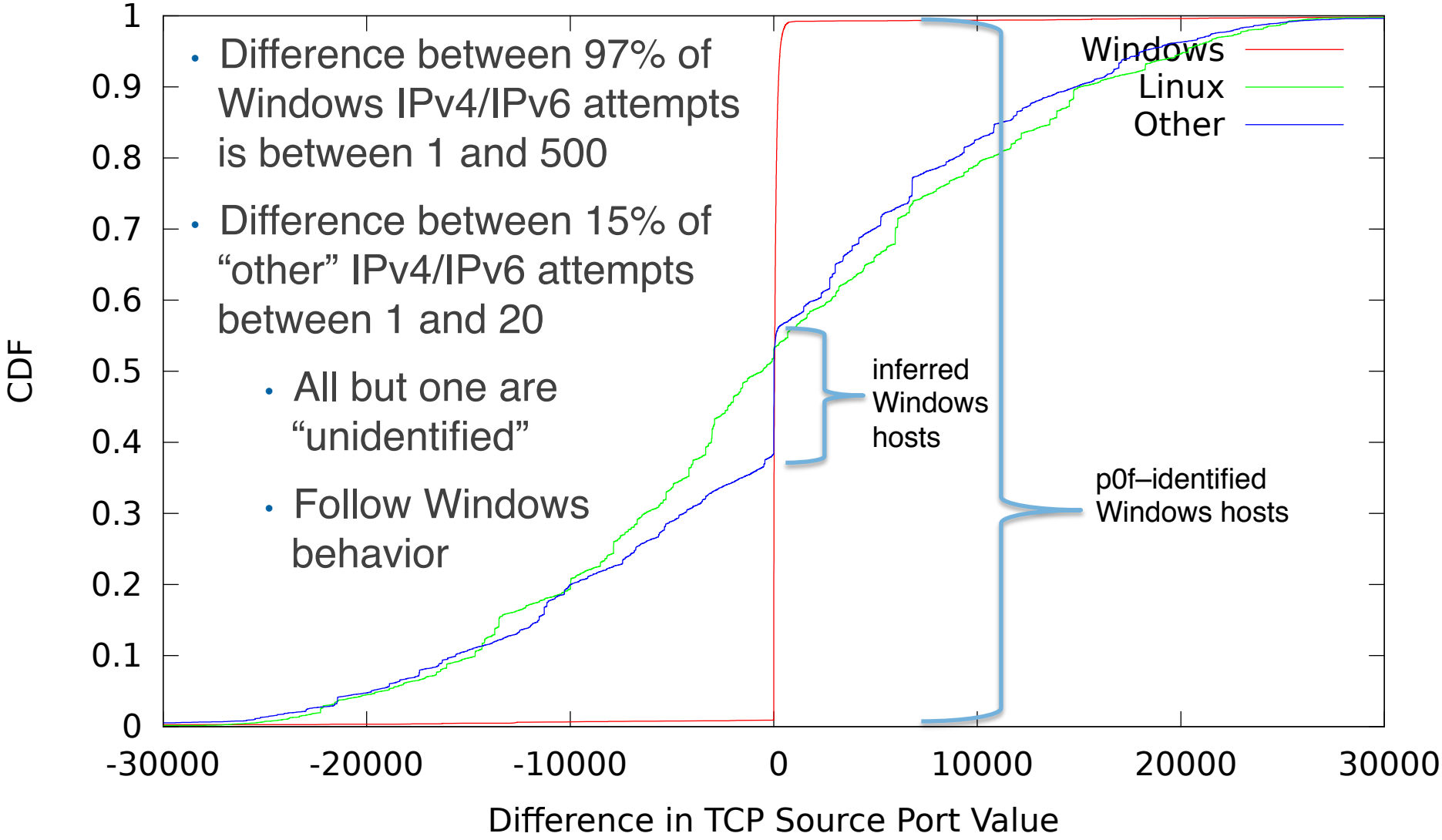
- Example – Windows XP/7

Sequential `connect()`	Source port
1. ::1	50673
2. 127.0.0.1	50674
3. ::1	50675
4. 127.0.0.1	50676

- Example – Linux

Sequential `connect()`	Source port
1. ::1	54382
2. 127.0.0.1	60164
3. ::1	54383
4. 127.0.0.1	60165

TCP Source Port Proximity Between IPv4/IPv6 Attempts



Misbehaving MTAs

- MTAs from Microsoft ASNs attempted over one million collective `connect()` attempts over one month (roughly one `connect()` every five seconds from each /64)

Subnet	Number of addresses	Attempts
2a01:111:f400:fe00::/64	4	538481
2a01:111:f400:fe04::/64	4	538174

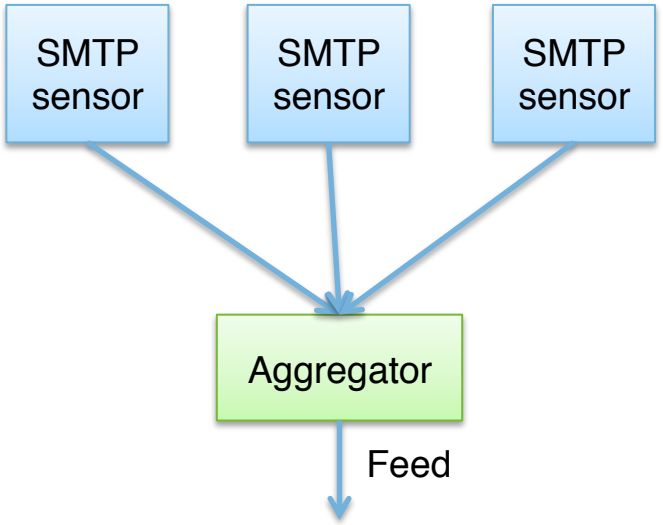
- Few corresponding IPv4 attempts during that time
- Apparently not associated with real attempts

Misbehaving MTAs – sendmail

- An instance of sendmail (v 8.13.8, distributed) issued requests from the same address and source port in succession over several weeks
- Few corresponding IPv4 attempts during that time
- Unable to reproduce this in an isolated lab environment
- Single `connect()` attempt or source port re-use?
- What caused this behavior?

Summary and Future Work

- **Summary**
  - Reputation of IPv6 Internet is largely unknown
  - Architecture for measuring abusive IPv6 SMTP on a production email domain has been presented
  - Moderate presence of spammers of various sources, though spam content can't be confirmed
- **Future work**
  - Further analyze existing data
  - Compare data with that of unused email domain
  - Create network of SMTP sensors, all contributing data (collaboration requested!)



powered by

VERISIGN™