

Filtering Spam with Behavioral Blacklisting

Anirudh Ramachandran, Nick Feamster, Santosh Vempala
College of Computing, Georgia Tech
`{avr, feamster, vempala}@cc.gatech.edu`

Spam volumes worse than ever

- Percentages of spam higher than ever (*the highest we've seen is 95% of email [1]*)
- Average sizes of spam message are increasing
 - Mail servers buckling under increased processing load
- *What are our options?*

Content-based filters

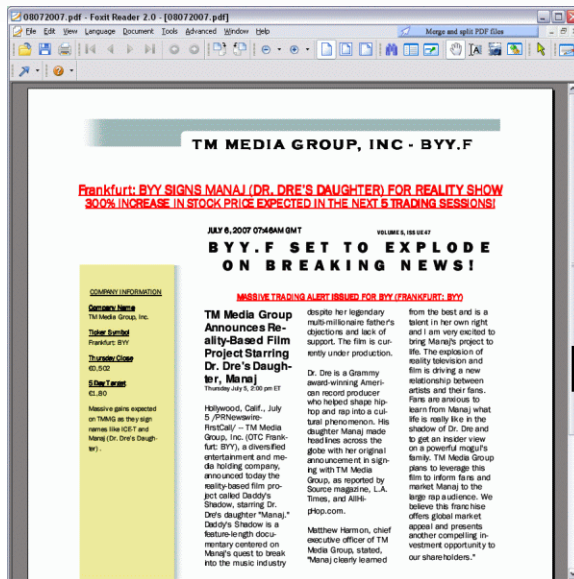
- Worked well in the age of text-only spam
- These days: *container* spam



Excel sheets

C	D	E	F	G	H	I	J	K
1								
2								
3								
4								
5								
6								
7								
8								
9								
10								
11								
12								
13								
14								
15								
16								

and even MP3s!



Content-based filters cannot keep up

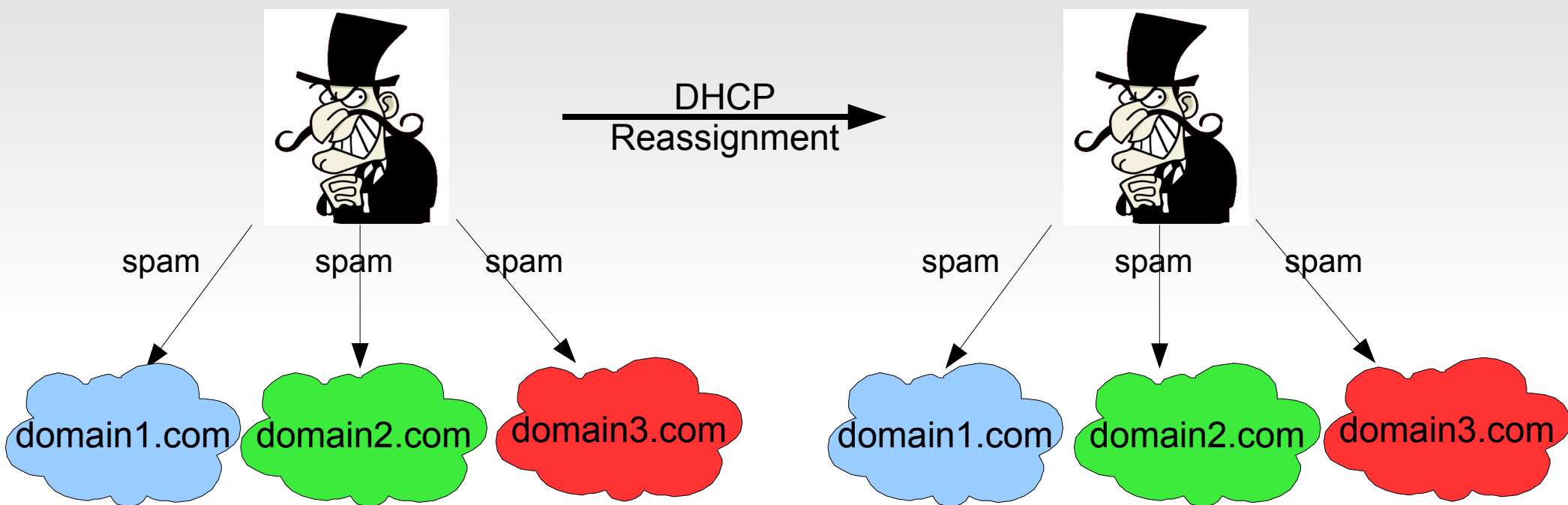
What about IP Blacklists?

- Cannot deal with sender *dynamism*
- Based on an *ephemeral identifier* (IP address)
 - Constant stream of new senders due to *renumbering* (DHCP), *infection*, or *prefix hijacking*
- Requires *human verification*

Why IP blacklists might fail

IP Address: **76.17.114.xxx**

IP Address: **24.99.146.xxx**



- IP Blacklists cannot make this connection
- *Spammer's sending pattern has not changed*

Blacklists: Incomplete and Unresponsive

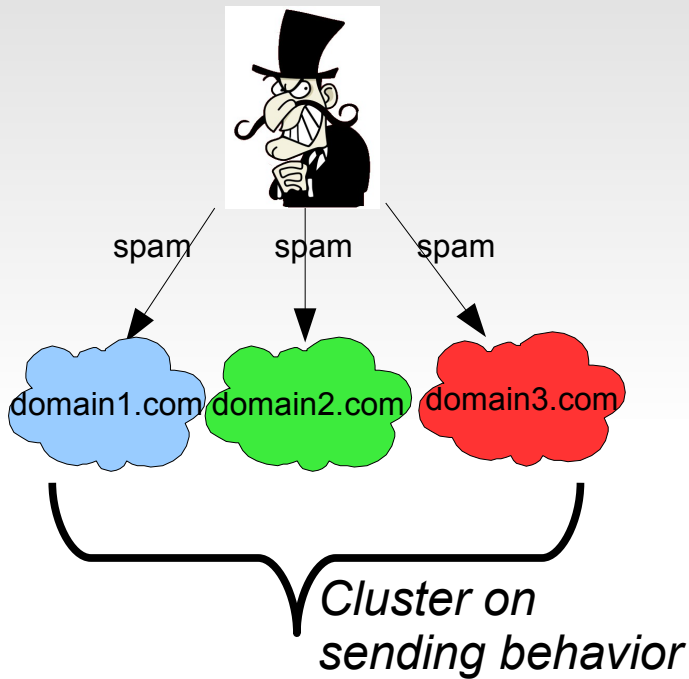
<i>Data Source</i>	<i>Spam</i>	<i>IPs</i>	<i>Spam from unlisted IPs</i>	
			<i>At Receipt</i>	<i>After 1 Month</i>
Trap 1	384,521	129,243	134,120 (35%)	79,532 (20%)
Trap 2	172,143	64,386	17,132 (10%)	14,534 (8.5%)

Unlisted spam senders during March 2007 at SpamHaus.org

- What about the ones that *do* get listed?
 - 10 – 15 % of *eventually listed* spammers take **over 30 days** to get listed!

How can SpamTracker help?

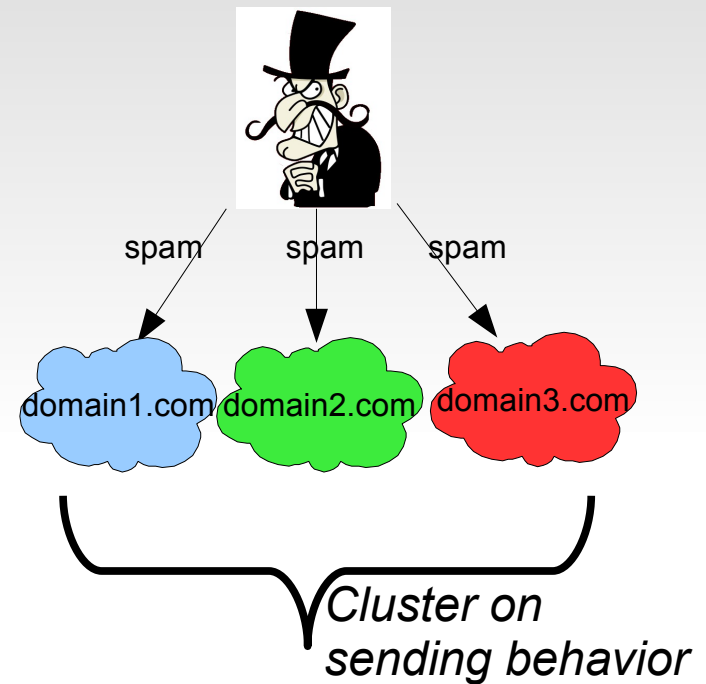
IP Address: **76.17.114.xxx**
Known Spammer



Domain No.	3	2	4	0	0
------------	---	---	---	---	---

Behavioral fingerprint

IP Address: **24.99.146.xxx**
Unknown sender



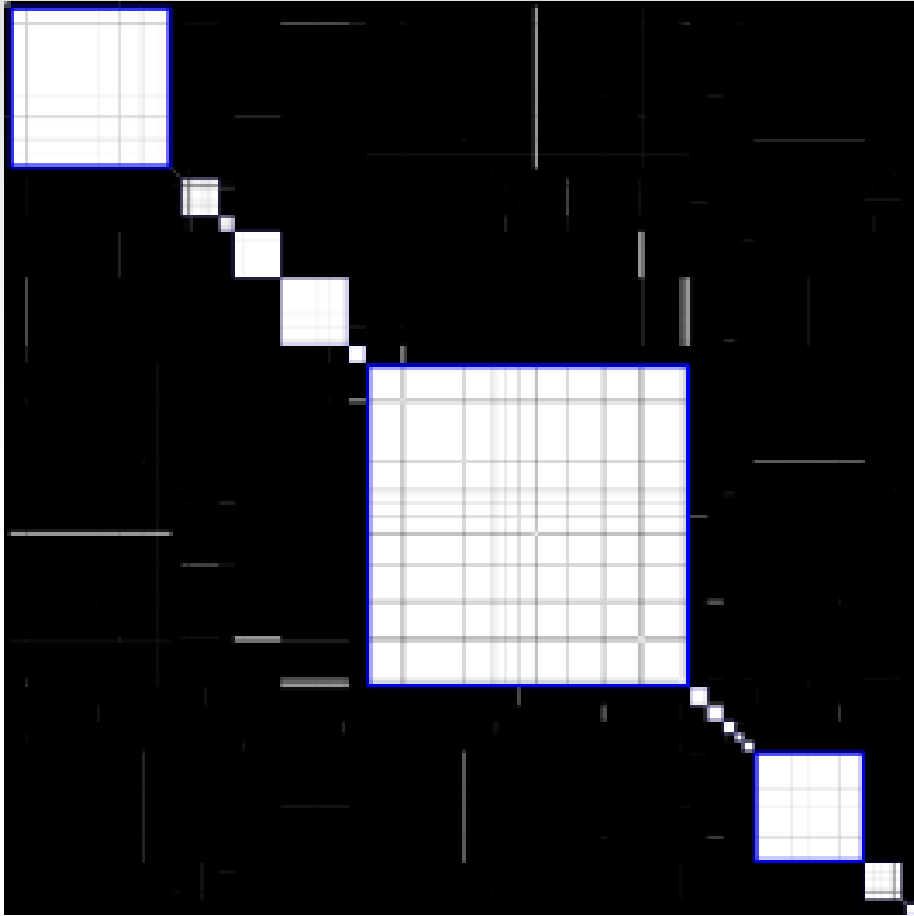
Domain No.	2	1	2	0	0
------------	---	---	---	---	---

Similar fingerprint!

SpamTracker: Idea

1. For each sender, *construct* a behavioral fingerprint
 - Based on domains targeted by sender
2. *Cluster* senders that have similar fingerprints
 - Clusters likely to comprise spammers
3. *Filter* new senders that map to existing clusters
 - Only spammers that have similar behavior to a cluster will map well on to it

Spamming behavior clusters well



Intensity(i,j) \sim Similarity(IP i , IP j)
Size: 2469x2469 (IPs x IPs)

- Each cluster denotes one sending pattern
- Very similar behavior within clusters (very different across)
- Legitimate senders do not form clusters
- Plausible explanation: *each cluster is one botnet*

Outline

- **Algorithm and Design**
 - Phase 1: Clustering
 - Phase 2: Classification
 - Design
- **Evaluation**
- **Future work and Deployment Options**

Phase 1: Clustering

- Steps in clustering
 - Collapse an *IP x domain x time* tensor to an *IP x domain* matrix (for, say, 6 hours) – $M(i, j)$
 - Group $M(i, j)$ into k clusters
 - $M_c(i, j)$ denotes the submatrix comprising rows of cluster c
 - c_{avg} denotes the *average row* of cluster c , *i.e.*,
$$c_{avg} = \frac{\sum_{i=1}^{|c|} M_c(i, j)}{|c|}$$
- c_{avg} can be pre-computed and stored for all clusters

Phase 2: Classification

- Steps in classification
 - For each new IP, construct a $1 \times d$ vector r encoding the IP's (recent) behavioral fingerprint
 - For each cluster c , compute similarity of r with c

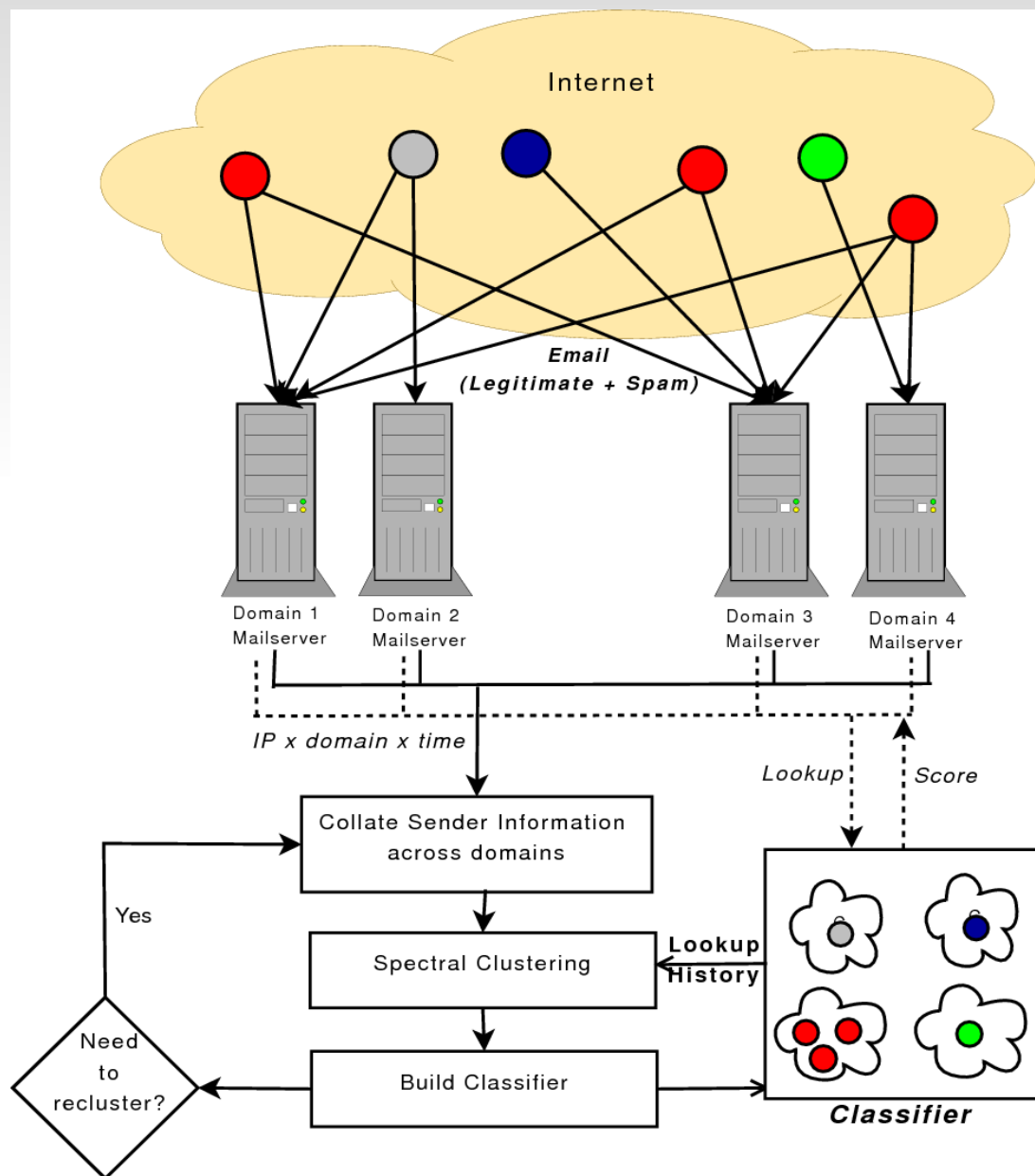
$$sim(r, c) = \frac{r \cdot c_{avg}}{|c_{avg}|} \quad (\text{normalized inner product})$$

- Compute the spam score $S(r)$ of the IP as

$$S(r) = \max_c sim(r, c) \quad (\text{maximum similarity to a cluster})$$

- $S(r)$ is used as a threshold to filter / greylist new senders

SpamTracker: Design



1. Collate sending IP, time and target domain info. for $[t, t + \Delta t)$
2. Cluster IPs according to set of domains targeted for $[t, t + \Delta t)$
3. Classify and simultaneously collect info for IPs in $[t + \Delta t, t + 2\Delta t)$

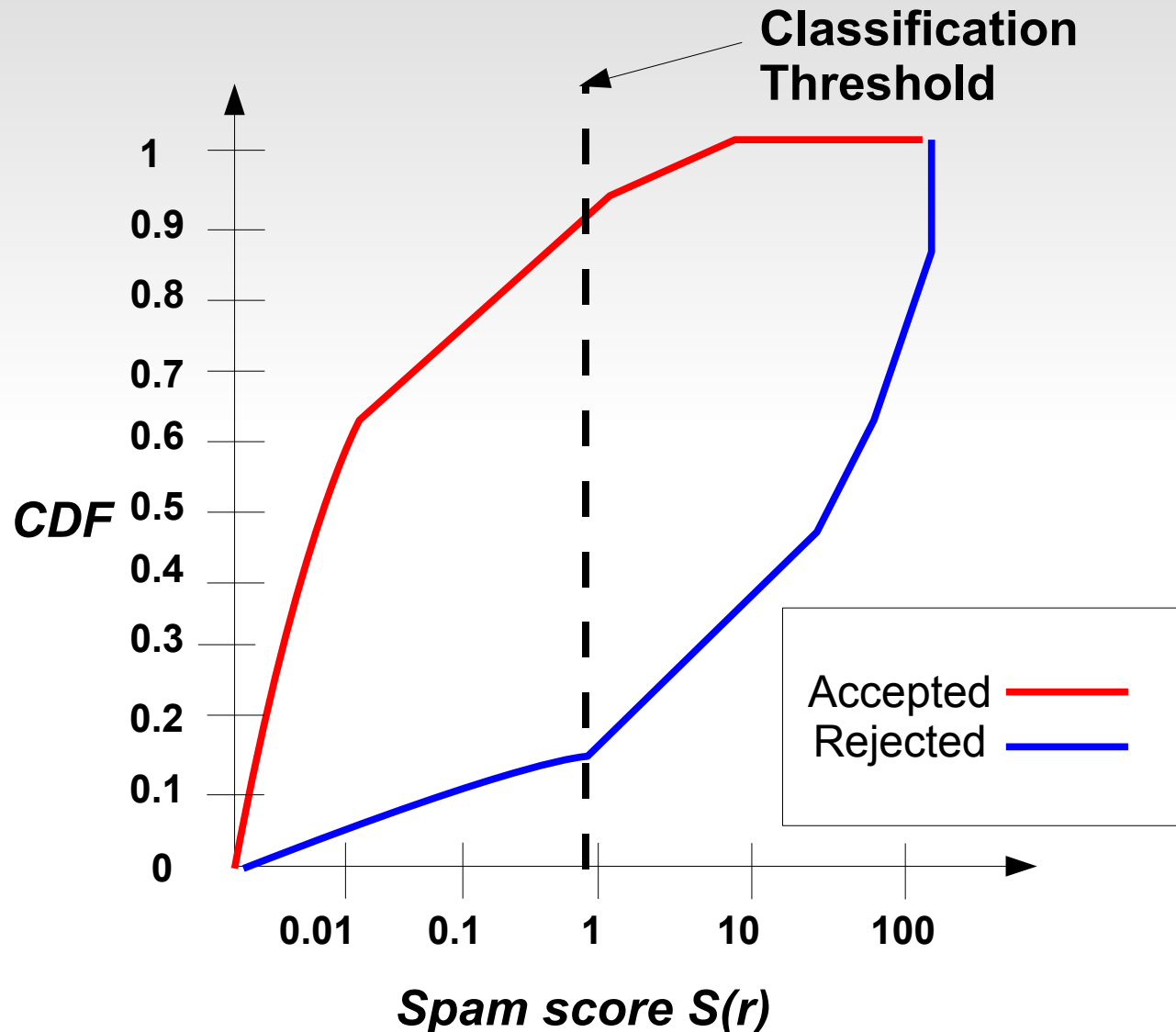
Outline

- Algorithm and Design
- **Evaluation**
- Future work and Deployment Options

Evaluation

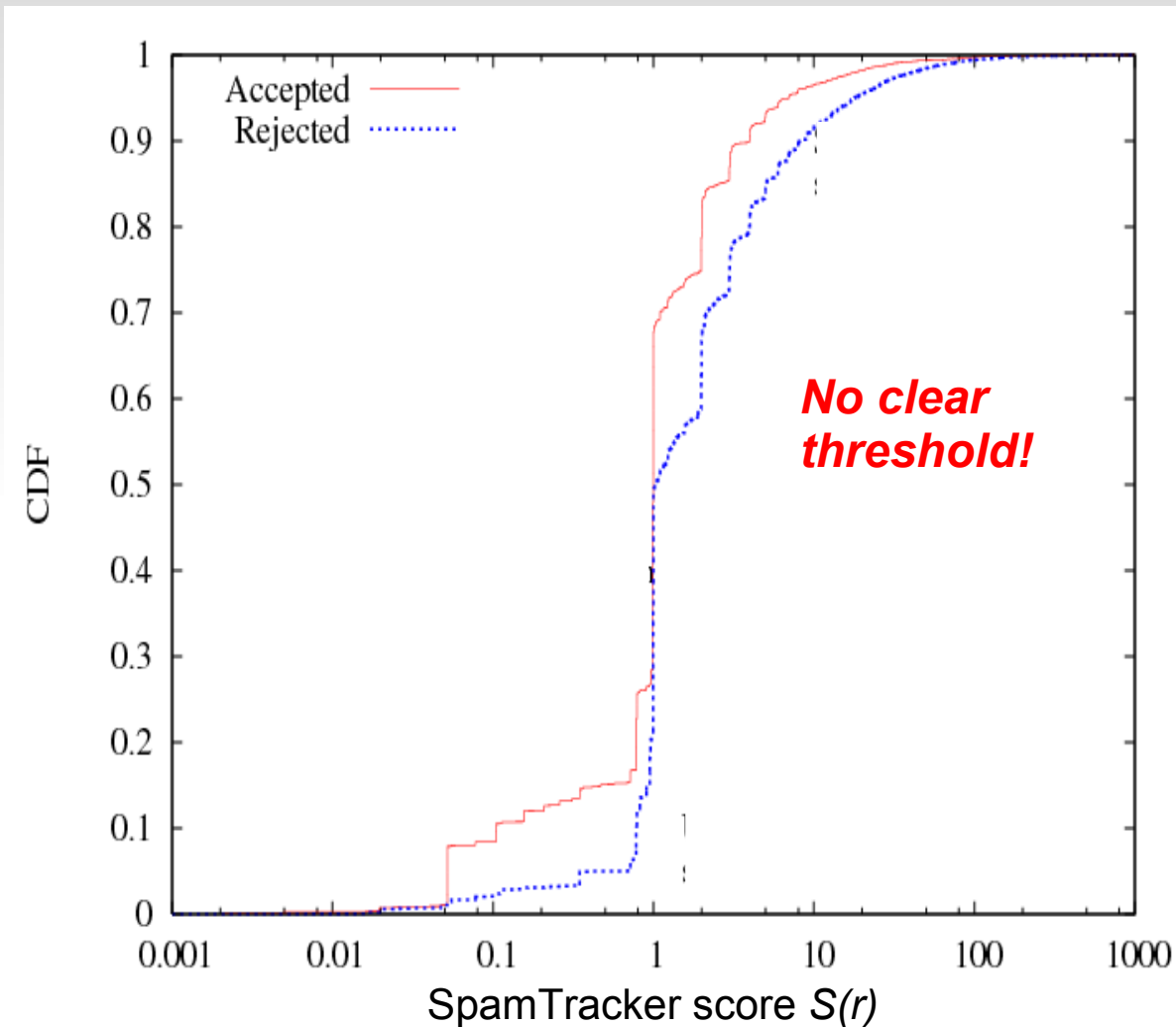
- Data
 - *Logs of legitimate and spam email* from an email service provider *for over 115 domains* for March 2007
 - Labeled (i.e., *accepted* or *rejected*) based on early drops
 - 40 million rejected, 5.5 million accepted
- Method
 - Compute clusters using *spam* senders in one interval
 - Use it to classify *all* senders in the next interval
 - $\Delta t = 6$ hour

What do we expect to see?



- A *clear threshold* to separate spammers from legitimate senders
- *i.e.*, SpamTracker's classification should (mostly) match the labels

What did we see?

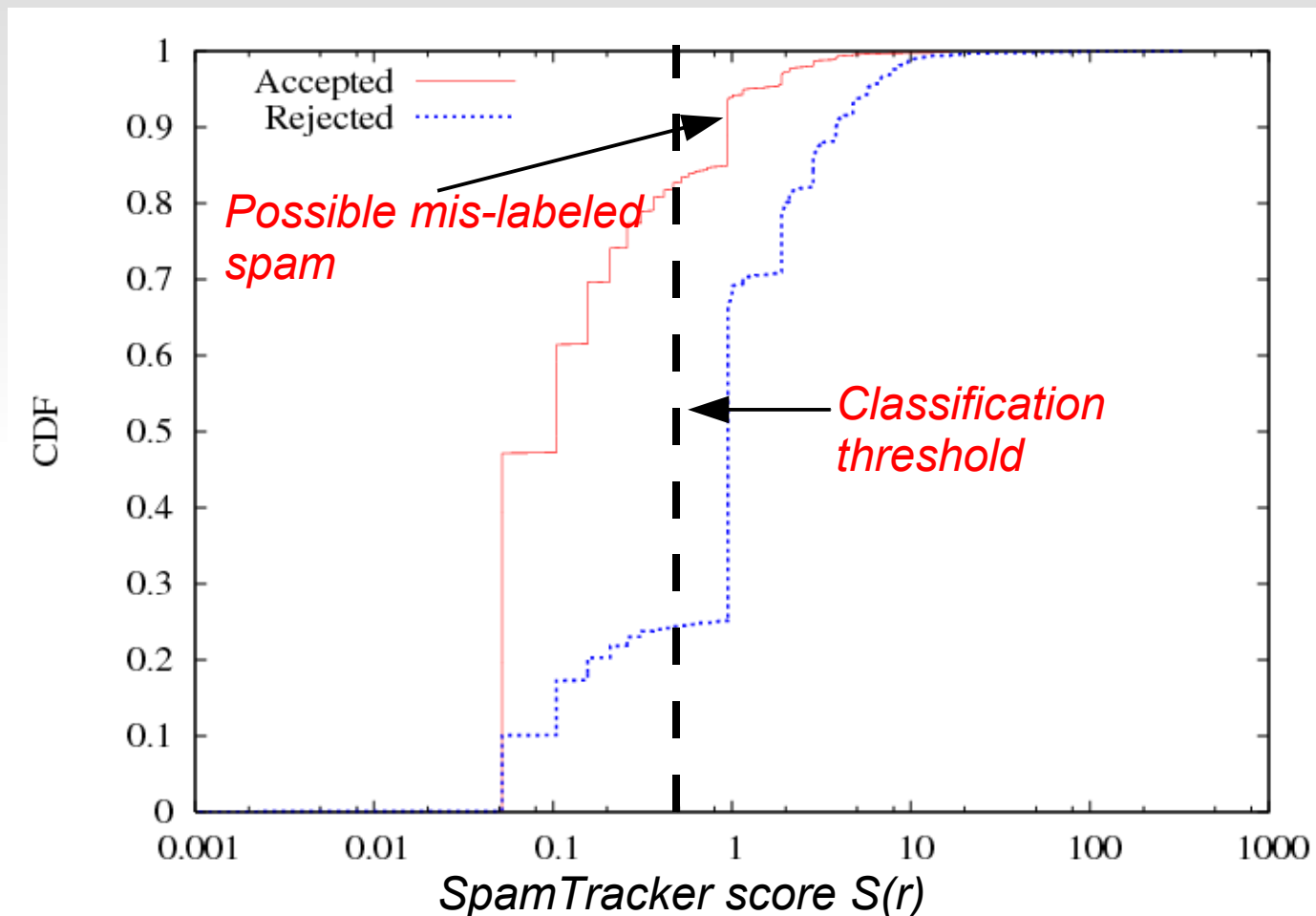


Spam score CDF for all IPs in a 6-hour interval

- Reason: *skewed data*
 1. Labels not accurate
 2. Most IPs sending to a few domains
- *Need to compare IPs that have multi-domain fingerprints*

Cluster with multi-domain fingerprint

Classification for IPs with maximum similarity to one cluster



Accepted IPs with high score appeared in blacklists later

Outline

- Algorithm and Design
- Evaluation
- **Discussion and Future Work**
- Deployment Options

Attacks against SpamTracker

- *Bots hit random domains from a list*
 - SpamTracker correctly clusters senders that pick targets from the *same distribution*
- *Bots try to emulate legitimate senders*
 - Bots may no longer be economically viable
 - Bots (programs) may not be able to send in true random fashion

Deployment Options

- *Option 1: Integration with existing infrastructure*
 - SpamTracker as another DNSBL
 - `dig 1.7.207.130.cc.gatech.edu.spamtracker.org`
 - Easy deployment; no changes at client-end
- *Option 2: On-the-wire deployment*
 - SpamTracker can get info by sniffing the wire
 - It only needs source IP address and destination domain
 - Wider view of spam, but harder deployment

Future Work

- ***Ongoing Improvements / Future Work***
 - ***Clustering on the time / frequency dimension (temporal clustering)***
 - Can identify phase-shifted senders with similar behavior
 - ***Improved similarity function:***
 - Weight domains that help differentiate senders

Summary

- *IP blacklists are ineffective* against a significant fraction of spam
- *Behavioral blacklisting* – new approach to spam filtering to augment IP blacklists
 - Filter based on *how* mail was sent, not *what* was sent
- SpamTracker is a promising behavioral blacklisting algorithm that filters based on *sending patterns*

Extra Slides

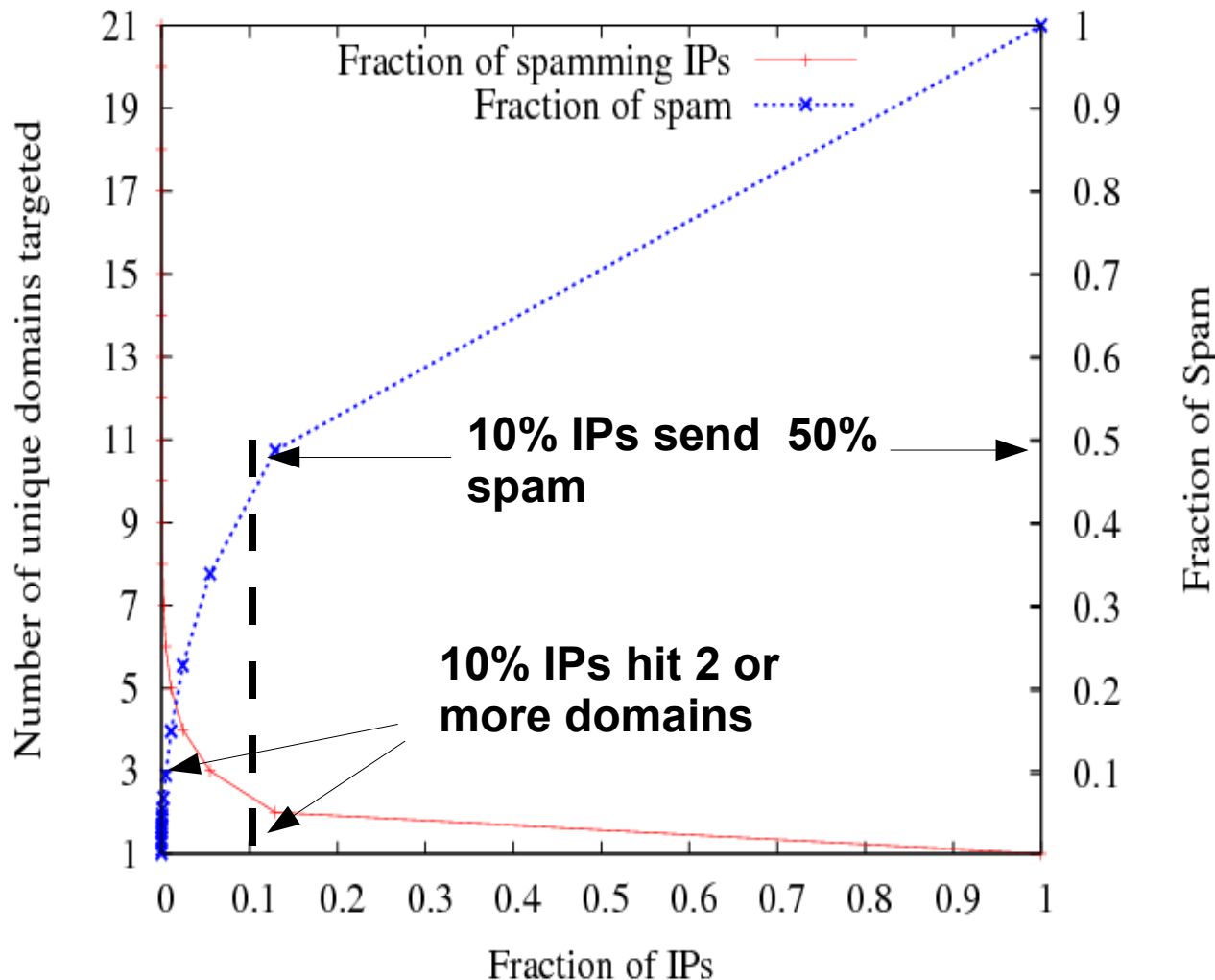
Spam fighting arsenal

- Existing solutions
 - Content filters
 - Does not work against attachment spam, does not scale
 - Heuristics (SPF, volume-based techniques, etc.)
 - Not a general solution...
 - IP / prefix blacklisting
 - Does not work against IP address dynamism
- Behavioral methods, *e.g.*, SpamTracker (*this paper*)
 - Filter based on *how* spam was sent, not what / who

The problem with IP blacklists

- Based on an ephemeral identifier (IP address)
 - Dynamic renumbering of IP addresses
 - Newly compromised machines
 - IP space hijacking
- Requires a human to report a spam sender
 - Delays in raising and processing reports
 - View of senders across domains neglected

Why Behavioral Blacklisting helps

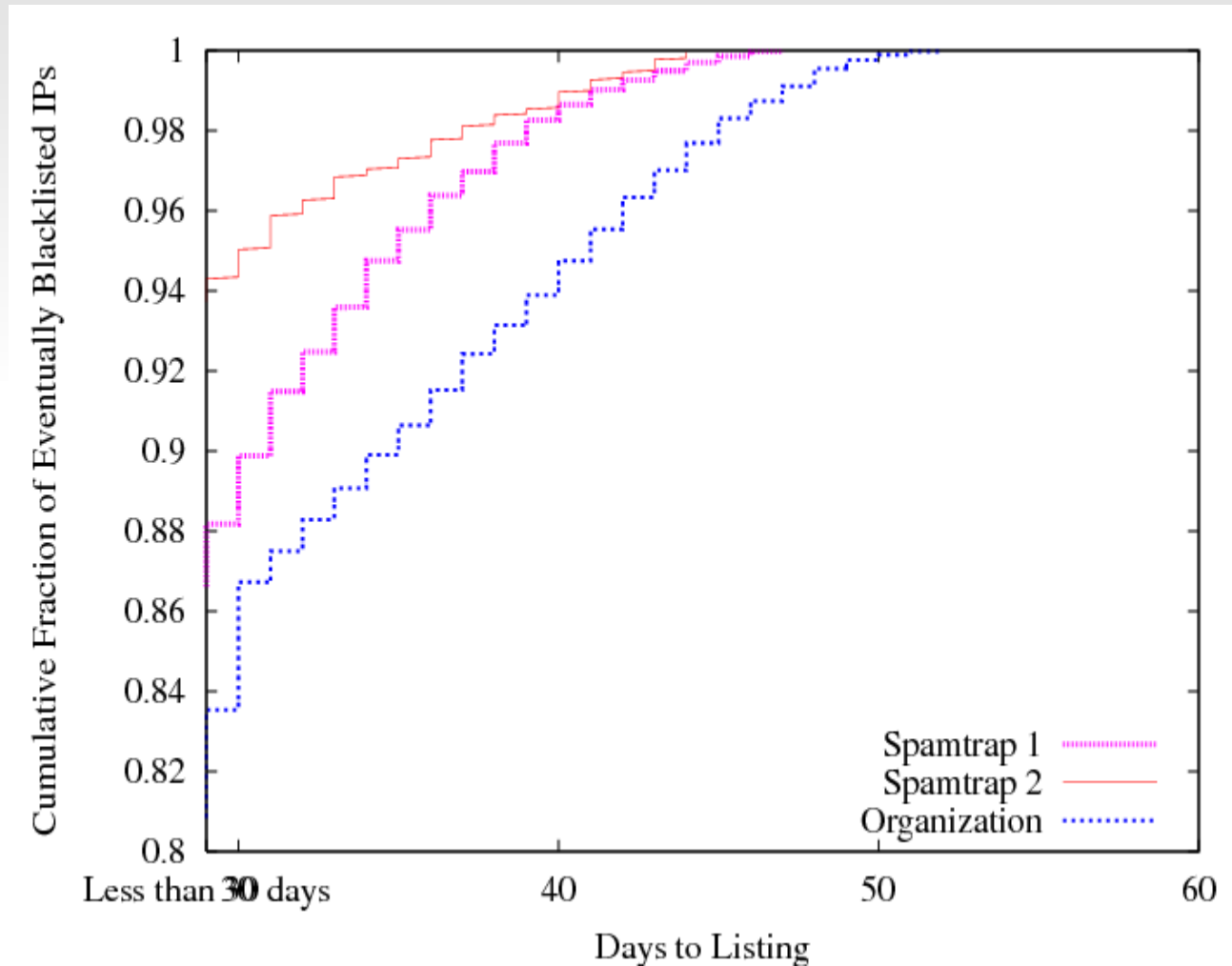


■ *Intuition*

- Attacks are distributed across many domains
- But, *senders in the same botnet are likely behave similarly*

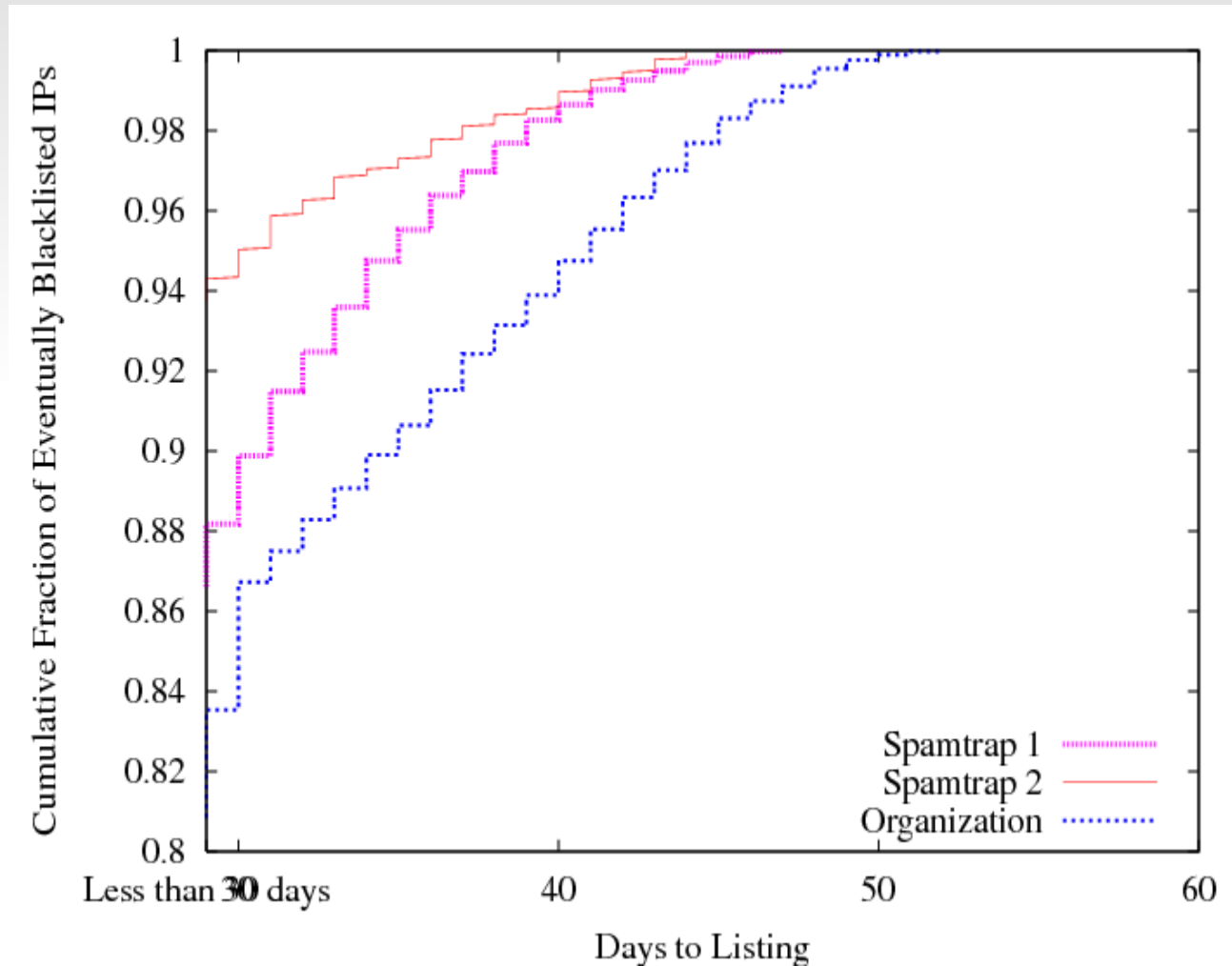
Blacklists: Unresponsive

- Time-to-listing of *unlisted* spammers that were *eventually listed* in SpamHaus
- **10 – 15 %** were listed only **30 days** after first spam was received



Blacklists: Unresponsive

- Time-to-listing of *unlisted* spammers that were *eventually listed* in SpamHaus
- **10 – 15 %** were listed only **30 days** after first spam was received



Complexity of Spectral Clustering

- $O(Mn \log n)$ for doc-term matrix with n objects and M non-zeroes.