Exploiting Transport-Level Characteristics of Spam

Robert Beverly\(^1\)  Karen Sollins

MIT Computer Science and Artificial Intelligence Laboratory

\(^1\)now at BBN Technologies

\{rbeverly,sollins\}@csail.mit.edu

August 21, 2008

Conference on Email and Anti-Spam 2008
Outline

1. Background
2. Experimental Methodology
3. Learning and Prediction
4. Open Questions
The Spam Arms Race

Attackers, scammers and thieves quickly adapt to defenses. Most effective solutions exploit *fundamental* weaknesses of attackers.

**Current Best Practices:**

- Content Filtering ... response: modify word tokens
- Reputation Analysis ... response: dynamic, fresh addresses
- Collaborative Filtering ... response: mail uniqueness
- And the cycle continues: Authentication Schemes, computational puzzles, etc.
The Spam Arms Race

Attackers, scammers and thieves quickly adapt to defenses. Most effective solutions exploit *fundamental* weaknesses of attackers.

**Current Best Practices:**
- Content Filtering ... response: modify word tokens
- Reputation Analysis ... response: dynamic, fresh addresses
- Collaborative Filtering ... response: mail uniqueness
- And the cycle continues: Authentication Schemes, computational puzzles, etc.
We propose a different approach:

- No panacea; existing solutions all have weaknesses
- Our solution, “SpamFlow,” is distinct from current practice

Question:
Are traffic characteristics a fundamental weakness of spam?
Specifically:

- What is the *transport* (TCP/IP packet stream) character of spam?
- Are there *differences* between spam and ham flows?
- How to exploit differences in a way which spammers cannot easily evade?

Why ask this question?
Hypothetical Question

Specifically:

- What is the *transport* (TCP/IP packet stream) character of spam?
- Are there *differences* between spam and ham flows?
- How to exploit differences in a way which spammers cannot easily evade?

Why ask this question?
Transport-Level Characteristics of Spam

Two Observations

1. Low Penetration:
   - due to existing filters, user ambivalence
   - → huge volumes of spam

2. Sending Methods:
   - Open mail relays, email trojans, botnets, dialup
   - → Low asymmetric bandwidth, widely distributed
Combining Observations: Low Penetration + Sending Methods

Volume + Methods + Economics $\rightarrow$ link/host resource contention

Contention:
Contention manifests as TCP/IP loss, retransmission, reordering, etc.
Understanding SpamFlow

- Not looking at IP header
- Not looking at data
- SpamFlow: TCP stream, including timing
- (look at combining methods later)
Outline

1. Background
2. Experimental Methodology
3. Learning and Prediction
4. Open Questions
Transmission Control Protocol (TCP):

- Reliable, bi-directional, in-order byte transmission abstraction
  - Acknowledgments
  - State Machine
- Flow and congestion control
  - Reacts to loss, persistent congestion
- Multi-flow fairness and efficient resource utilization (AIMD)
  - Round trip time (RTT) estimation
  - Bandwidth probing
Transmission Control Protocol:

- Simple Mail Transport Protocol (SMTP) uses TCP for transport
- Sequence of SMTP handshaking between Mail Transport Agents (MTAs)
- Mail contents are packetized
Outline

1. Background
2. Experimental Methodology
3. Learning and Prediction
4. Open Questions
### How do Spam Connections Behave?

...or, a quick look at `netstat`

<table>
<thead>
<tr>
<th>RcvQ</th>
<th>SndQ</th>
<th>Local</th>
<th>Foreign Addr</th>
<th>State</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>srv:25</td>
<td>92.47.129.89:49014</td>
<td>SYN_RECV</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>srv:25</td>
<td>ppp83-237-106-114...:29081</td>
<td>SYN_RECV</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>srv:25</td>
<td>88.200.227.123:25068</td>
<td>SYN_RECV</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>srv:25</td>
<td>92.47.129.89:49014</td>
<td>SYN_RECV</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>srv:25</td>
<td>ppp83-237-106-114...:29084</td>
<td>SYN_RECV</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>srv:25</td>
<td>88.200.227.123:25068</td>
<td>SYN_RECV</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>srv:25</td>
<td>88.200.227.123:25069</td>
<td>SYN_RECV</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>srv:25</td>
<td>88.200.227.123:25070</td>
<td>SYN_RECV</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>srv:25</td>
<td>88.200.227.123:25074</td>
<td>SYN_RECV</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>srv:25</td>
<td>84.255.150.15:4232</td>
<td>SYN_RECV</td>
</tr>
<tr>
<td>25</td>
<td></td>
<td>srv:25</td>
<td>222.123.147.41:50282</td>
<td>LAST_ACK</td>
</tr>
<tr>
<td>28</td>
<td></td>
<td>srv:25</td>
<td>adsl-pool-222.123...:1720</td>
<td>LAST_ACK</td>
</tr>
<tr>
<td>31</td>
<td></td>
<td>srv:25</td>
<td>222.123.147.41:50152</td>
<td>LAST_ACK</td>
</tr>
<tr>
<td>15</td>
<td></td>
<td>srv:25</td>
<td>222.123.147.41:50889</td>
<td>LAST_ACK</td>
</tr>
<tr>
<td>9</td>
<td></td>
<td>srv:25</td>
<td>88.245.3.19:venus</td>
<td>LAST_ACK</td>
</tr>
<tr>
<td>25</td>
<td></td>
<td>srv:25</td>
<td>78.184.155.70:1854</td>
<td>FIN_WAIT1</td>
</tr>
<tr>
<td>23</td>
<td></td>
<td>srv:25</td>
<td>190-48-30-225.spe:50920</td>
<td>FIN_WAIT1</td>
</tr>
<tr>
<td>23</td>
<td></td>
<td>srv:25</td>
<td>dsl.dynamic812132:48154</td>
<td>FIN_WAIT1</td>
</tr>
<tr>
<td>23</td>
<td></td>
<td>srv:25</td>
<td>ip-85-160-91-16.e:48093</td>
<td>FIN_WAIT1</td>
</tr>
<tr>
<td>23</td>
<td></td>
<td>srv:25</td>
<td>88.234.141.158:48389</td>
<td>FIN_WAIT1</td>
</tr>
<tr>
<td>23</td>
<td></td>
<td>srv:25</td>
<td>p5B0FBB5D.dip.t-d:11965</td>
<td>FIN_WAIT1</td>
</tr>
</tbody>
</table>

...
### How do Spam Connections Behave?

...or, a quick look at `netstat`

<table>
<thead>
<tr>
<th>RcvQ</th>
<th>SndQ</th>
<th>Local</th>
<th>Foreign Addr</th>
<th>State</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>srv:25</td>
<td>92.47.129.89:49014</td>
<td>SYN_RECV</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>srv:25</td>
<td>ppp83-237-106-114:29081</td>
<td>SYN_RECV</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>srv:25</td>
<td>88.200.227.123:25068</td>
<td>SYN_RECV</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>srv:25</td>
<td>92.47.129.89:49014</td>
<td>SYN_RECV</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>srv:25</td>
<td>ppp83-237-106-114:29084</td>
<td>SYN_RECV</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>srv:25</td>
<td>88.200.227.123:25068</td>
<td>SYN_RECV</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>srv:25</td>
<td>88.200.227.123:25069</td>
<td>SYN_RECV</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>srv:25</td>
<td>88.200.227.123:25070</td>
<td>SYN_RECV</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>srv:25</td>
<td>84.255.1.127:2301</td>
<td>SYN_RECV</td>
</tr>
<tr>
<td>0</td>
<td>25</td>
<td>srv:25</td>
<td>222.123.147.41:50282</td>
<td>LAST_ACK</td>
</tr>
<tr>
<td>0</td>
<td>28</td>
<td>srv:25</td>
<td>adsl-pool-74-40:1720</td>
<td>LAST_ACK</td>
</tr>
<tr>
<td>0</td>
<td>31</td>
<td>srv:25</td>
<td>222.123.147.41:50152</td>
<td>LAST_ACK</td>
</tr>
<tr>
<td>0</td>
<td>15</td>
<td>srv:25</td>
<td>222.123.147.41:50889</td>
<td>LAST_ACK</td>
</tr>
<tr>
<td>0</td>
<td>9</td>
<td>srv:25</td>
<td>88.245.3.19:venus</td>
<td>LAST_ACK</td>
</tr>
<tr>
<td>0</td>
<td>25</td>
<td>srv:25</td>
<td>78.184.1.127:2301</td>
<td>FIN_WAIT1</td>
</tr>
<tr>
<td>0</td>
<td>23</td>
<td>srv:25</td>
<td>190-48-30-225.spe:50920</td>
<td>FIN_WAIT1</td>
</tr>
<tr>
<td>0</td>
<td>23</td>
<td>srv:25</td>
<td>dsl.dynamic812132:48154</td>
<td>FIN_WAIT1</td>
</tr>
<tr>
<td>0</td>
<td>23</td>
<td>srv:25</td>
<td>ip-85-160-91-16.e:48093</td>
<td>FIN_WAIT1</td>
</tr>
<tr>
<td>0</td>
<td>23</td>
<td>srv:25</td>
<td>88.234.141.158:48389</td>
<td>FIN_WAIT1</td>
</tr>
<tr>
<td>0</td>
<td>23</td>
<td>srv:25</td>
<td>p5B0FBB5D.dip.t-d:11965</td>
<td>FIN_WAIT1</td>
</tr>
</tbody>
</table>

**TCP Stuck in States**

- Stays in these states for minutes
- Half-open connections
- Remote MTAs that “disappear” mid-connection
- Remote MTAs that send `FIN` and disappear

R. Beverly, K. Sollins (MIT)
What about RTT?
...building more intuition

Dear Robert,
I hope you have had a great week!

A Tag Heuer watch is a luxury statement on its own.
In Prestige Replicas, any Tag Heuer...
**Data Collection**

- Instrument a Mail Transport Agent (MTA) server
- Collect SMTP packet trace
- Match labeled emails to packet flows

**Diagram:**

- TCP/IP
- MTA
- Mail
- SMTP Packet Capture
- Flows
- Match
- Labels
- Spam/Ham?
- Dataset \((\bar{X}, \bar{Y})\)
Outline

1. Background
2. Experimental Methodology
3. Learning and Prediction
4. Open Questions
Round Trip Time

Experimental Methodology

Using a flow property

Round Trip Time

Cumulative Probability

RTT (sec)

Spam
Ham

P(\text{ham rtt} < 100\text{ms}) \sim 1;
P(\text{spam rtt} < 100\text{ms}) \sim 0.2!
Bayes’ Rule

- Use causal information to form diagnosis

\[ P(spam|rtt > x) = \frac{P(rtt > x|spam)P(spam)}{P(rtt > x)} \]  

(1)
Round Trip Time

cont'd

Neutral between [20 – 100 ms]; Highly biased otherwise
Experimental Methodology

Using a flow property

Selecting Features

Wait! You’re disenfranchising distant servers!
- Yes; may be a good thing
- $\sim 5\% > 1s$
- More importantly...

Other Transport “Features:”
- Packets, Retransmits, OutOfOrder, RSTs, FINs
- Zero Window, Minimum Cong. Window, Max Idle, Jitter, etc.
- *Adaptable* per-user, per-network

Key Insight

Statistical flow properties can provide differentiation
### Selecting Features

#### Wait! You’re disenfranchising distant servers!
- Yes; may be a good thing
- $\approx 5\% > 1\text{s}$
- More importantly...

#### Other Transport “Features:”
- Packets, Retransmits, OutOfOrder, RSTs, FINs
- Zero Window, Minimum Cong. Window, Max Idle, Jitter, etc.
- *Adaptable* per-user, per-network

---

### Key Insight

Statistical flow properties can provide differentiation
Outline

1. Background
2. Experimental Methodology
3. Learning and Prediction
4. Open Questions
Non-Features

- Many intuitively “good” features turn out not to be
- Strength of statistical approach

One Example in Detail:

- RSTs as abortive close on socket
- A good indication of misbehaving flows?
Non-Features

Example: Received RSTs

- Only $\sim 50\%$ of ham flows sent no RSTs!
- $\sim 30\%$ of ham flows send two RSTs!
- (see tech report for why)
Outline

1 Background

2 Experimental Methodology

3 Learning and Prediction

4 Open Questions
Picking Features

So, which features provide discrimination?

- Feature selection
- Simple method is forward fitting
- Greedily choose one available feature to minimize training error

\[ \hat{y} = \theta_0 + \sum \theta_i x_i \]

\[ \begin{bmatrix}
    1 & 0 & 1 & 0 \\
    -1 & 1 & 1 & 1 \\
    -1 & 1 & 0 & 0 \\
    \vdots & \vdots & \vdots & \vdots \\
    \end{bmatrix} \quad \begin{bmatrix}
    \vdots \\
    \hat{X} \\
    \vdots \\
    \end{bmatrix} \quad \begin{bmatrix}
    0 \\
    1 \\
    1 \\
    \vdots \\
    \end{bmatrix} \]

R. Beverly, K. Sollins (MIT)
Picking Features cont’d

80% chance that RTT or CwndMin is best single feature
Experimental Methodology

Feature Selection

Features

cont’d

<table>
<thead>
<tr>
<th>Feature</th>
<th>Selection Order</th>
</tr>
</thead>
<tbody>
<tr>
<td>CwndMin</td>
<td></td>
</tr>
<tr>
<td>RecvRxmit</td>
<td></td>
</tr>
<tr>
<td>RTT</td>
<td></td>
</tr>
<tr>
<td>RecvFIN</td>
<td></td>
</tr>
<tr>
<td>SentFIN</td>
<td></td>
</tr>
<tr>
<td>SentRxmit</td>
<td></td>
</tr>
<tr>
<td>SentRST</td>
<td></td>
</tr>
<tr>
<td>Cwnd0</td>
<td></td>
</tr>
</tbody>
</table>

Less discriminatory secondary features
Features

cont’d

The graph shows the probability density function (PDF) of various features sorted by their selection order. The features include:

- CwndMin
- RecvRxmit
- RTT
- RecvFIN
- SentFIN
- SentRxmit
- SentRST
- Cwnd0
- RecvPkt
- SentPkt
-RecvRST
-MaxIdle

The x-axis represents the selection order, while the y-axis shows the PDF values. Each feature is represented by a line in a different color, indicating its distribution across different selection orders.
Outline

1. Background
2. Experimental Methodology
3. Learning and Prediction
4. Open Questions
SpamFlow

Based on observations, build a model

- Supervised learning, binary classification
- E.g. Bayes Nets, Support Vector Machines, etc.

SpamFlow

- A working implementation of the ideas using SVMs

Evaluation

- $FP = \text{ham marked as spam}$
- $FN = \text{spam marked as ham}$
- $accuracy = \frac{TP + TN}{P + N}$
- $precision = \frac{TP}{TP + FP}$
Prediction Performance

- Multiple independent experiments
- Over $\sim 90\%$ accuracy, precision and recall
- Tight bounds
SpamAssassin False Negatives

- Against our data set, SpamAssassin gives 127 false negatives.
- SpamFlow detects 78% of those.
- → useful to combine methods!

For example...
SpamAssassin False Negatives

Re" Your Pharmacy order # 85493899

Pls Go ’ www.protectfair ’ dot com
SpamAssassin False Negatives

Received: (qmail 12851 invoked from network); 24 Jan 2008 05:14:58 -0000
Received: from 201-213-46-215.net.prima.net.ar (201.213.46.215:8963) 
by ralph.rbeverly.net with SMTP; 24 Jan 2008 05:14:58 -0000
Received: from unknown (HELO deviant) (192.168.0.5) by mail6.colossal.com 
with SMTP; Thu, 24 Jan 2008 00:14:58 -0500
Date: Thu, 24 Jan 2008 00:14:58 -0500
To: rbeverly@grdata.com, rcmsjm@grdata.com, reb3@grdata.com, 
roots.nojunk@grdata.com, russell_shute@grdata.com,
From: "Jordan Abrams" <inclusionVito@familyhistree.com>
Subject: Canadian Pharmacy Online! – 70-80% OFF!
Content-Length: 76
Lines: 6

Re" Your Pharmacy order # 85493899

Pls Go ' www.protectfair' dot com

SpamAssassin:
X-Spam-Status: No,
score=3.5 required=5.0
tests=BAYES_50,
FS_OBFU_PRMCY,
SORTED_RECIPS,
UNPARSEABLE_RELAY
autolearn=no version=3.2.3
SpamAssassin False Negatives

Received: (qmail 12851 invoked from network); 24 Jan 2008 05:14:58 -0000
Received: from 201-213-46-215.net.prima.net.ar (201.213.46.215:8963)
by ralph.rbeverly.net with SMTP; 24 Jan 2008 05:14:58 -0000
Received: from unknown (HELO deviant) (192.168.0.5) by mail6.colossal.com
with SMTP; Thu, 24 Jan 2008 00:14:58 -0500
Date: Thu, 24 Jan 2008 00:14:58 -0500
To: rbeverly@grdata.com, rcmsjm@grdata.com, reb3@grdata.com,
    roots.nojunk@grdata.com, russell_shute@grdata.com,
From: "Jordan Abrams" <inclusionVito@familyhistree.com>
Subject: Canadian Pharmacy Online! - 70-80% OFF
Content-Length: 76
Lines: 6

Re" Your Pharmacy order # 85493899
Pls Go ' www.protectfair ' dot com

SpamFlow:
SntPkt: 45 RcvPkt: 29
SntRxmit: 0 RcvRxmit: 1
SntRST: 0 RcvRST: 0
SntFIN: 1 RcvFIN: 1
Cwnd0: 0 MinCwnd: 65280
MaxIdle: 1.366636
RTT: 0.162413
Open Questions

Spam is an Arms Race:
- How would spammers react?
- Adapt by slowing down, sending less mail
- Could spammers tweak TCP stacks and circumvent?

Future Work:
- Gather additional data sets
- Package, distribute
- Explore method’s potential in other domains
Attacking spam at a different layer
Correct predictions with over 90% accuracy, precision and recall without content or reputation analysis
SpamFlow finds 78% of SpamAssassin false-negatives
No implementation hurdle, easily combined with existing techniques

Thanks!

Questions?
Can SpamFlow be more conservative in using RTT: Yes, even a highly conservative filter can still leverage RTT to eliminate extremely large RTT spam flows.

Doesn’t SpamFlow privilege well-connected senders? Personal, home or small business servers do not have the same volume requirement as spammers and thus are unlikely to induce the same TCP congestion effects we observe. SpamFlow only discriminates against sources that are both poorly connected and injecting large volumes of mail.

What about email lists? In contrast to spam, which must be sent continually, email list traffic can be scheduled in order to not cause local congestion.
Support Vector Machines

Dual-Form, Constrained Optimization:

\[
\sum_{t=1}^{n} \alpha_t - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j K(\phi(x_i), \phi(x_j)) \quad \text{s.t.} \quad C \geq \alpha_t \geq 0, \quad \sum_{t=1}^{n} \alpha_t y_t = 0 \quad (2)
\]

- Separate training set into two classes in most general way
- **Main insight**: find hyper-plane separator that maximizes the minimum margin between convex hulls of classes
- **Second insight**: if data is not linearly separable, take to higher dimension
- **Result**: generalizes well, fast, accommodate unknown data structure
What’s going on here?
Example: Received RSTs

---

Google sends SMTP **QUIT**, then active close, then **RSTs** passive close

```
11:55:57.807504 googl > srv: P 187089:187095(6) ack 143 win 5720
11:55:57.807510 googl > srv: F 187095:187095(0) ack 143 win 5720
11:55:57.807628 srv > googl: .  ack 187096 win 32614
11:55:57.808181 srv > googl: F 167:167(0) ack 187096 win 32614
11:55:57.834759 googl > srv: R 46149836:46149836(0) win 0
```

Yahoo! sends SMTP **QUIT**, srv performs active close. Yahoo! then sends three **RSTs** when srv goes to **TIME_WAIT**

```
11:20:35.023406 srv > yahoo: P 113:137(24) ack 1426 win 32120
11:20:35.023782 srv > yahoo: F 137:137(0) ack 1426 win 32120
11:20:35.023983 yahoo > srv: F 1426:1426(0) ack 113 win 33304
11:20:35.024073 srv > yahoo: .  ack 1427 win 32120
11:20:35.076591 yahoo > srv: R 776208340:776208340(0) win 0
11:20:35.076969 yahoo > srv: R 776208340:776208340(0) win 0
11:20:35.077381 yahoo > srv: R 776208341:776208341(0) win 0
```

Abortive close in Postfix source; normal behavior
What’s going on here?
Example: Received RSTs

Is abortive close a common “normal” SMTP technique?

Postfix Source

```c
static void start_connect(SESSION *session) {
    int fd;
    struct linger linger;
    linger.l_onoff = 1;
    linger.llinger = 0;
    if (setsockopt(fd, SOL_SOCKET, SO_LINGER, (char *) &linger,
                   sizeof(linger)) < 0)
        ...
```
Larger training sizes perform better

In practice: SpamFlow as a weighted voter
Features cont’d

- 80% chance that RTT or CwndMin is best first feature
- Others (e.g. RecvRST) much less discriminatory
Data Collection

Dataset:
- One week, January 2008
- $\sim 18k$ emails, only $\sim 200$ legitimate ham
- Normalize spam and ham count for each experiment, randomly select spams
- Dataset is small; future work corrects this

This talk: method, intuition, validation