



2011-10-03

# Transport-Layer Abusive Traffic Detection and Mitigation

Beverly, Robert

Monterey, California: Naval Postgraduate School.

---

<http://hdl.handle.net/10945/41917>



Calhoun is a project of the Dudley Knox Library at NPS, furthering the precepts and goals of open government and government transparency. All information contained herein has been approved for release by the NPS Public Affairs Officer.

**Dudley Knox Library / Naval Postgraduate School  
411 Dyer Road / 1 University Circle  
Monterey, California USA 93943**

<http://www.nps.edu/library>

# Transport-Layer Abusive Traffic Detection and Mitigation

Robert Beverly, Georgios Kakavelakis, Le Nolan, Joel Young

Center for Measurement and Analysis of Network Data  
Naval Postgraduate School, Dept. Computer Science  
{rbeverly,gkakavel,lenolan,jdyoung}@nps.edu  
October 3, 2011

ITACS CENIC Meeting 2011



# Outline

- 1 Background
- 2 Detecting Bot-Generated Spam
- 3 Real-world Botnet Detection
- 4 Current Research



# Internet Abusive Traffic

## Abusive traffic abounds on the Internet:

- e.g. email, phishing, malware, DoS, CAPTCHA solvers, etc.
- *Botnets* are a significant source of abusive traffic
- Large potential for damage
- Botnets becoming increasingly sophisticated (motivated economically, politically, militarily)
- e.g. distributed C&C, layers of obfuscation, re/mis-direction, etc.



# Botnet Arms Race

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

## Some Current Approaches:

- Reputation (e.g. blacklist) ... response: dynamic, fresh addresses
- Attack signatures ... response: polymorphism, etc.
- C&C signatures ... response: distributed C&C, encryption, etc.
- Communication structure of C&C ... response: mimic humans



# Botnet Arms Race

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

## Some Current Approaches:

- Reputation (e.g. blacklist) ... response: dynamic, fresh addresses
- Attack signatures ... response: polymorphism, etc.
- C&C signatures ... response: distributed C&C, encryption, etc.
- Communication structure of C&C ... response: mimic humans



# Our Research

## Transport-level (e.g. TCP) traffic signal analysis:

- Distinct from current practice and research ( $\neq$  Netflow analysis)
- *Key insight*: local botnet behavior manifests remotely as discriminative signal
- *Exploit lowest-level dependence*: sourcing large amounts of data (whether for spam, scam-hosting, attacks, etc).

Funded in part by: Cisco University Research Grant and the NSF.  
Thanks to NPS ITACS for supporting this research.



# Outline

- 1 Background
- 2 Detecting Bot-Generated Spam**
- 3 Real-world Botnet Detection
- 4 Current Research





# 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?



# 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

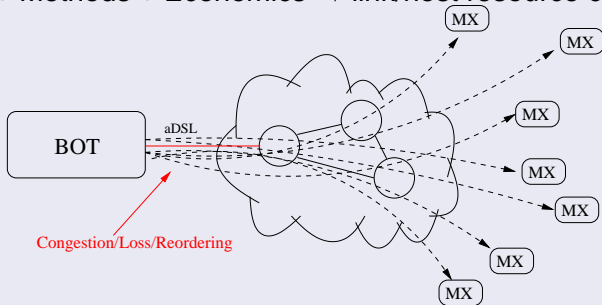
- 1 Low Penetration:
  - due to existing filters, user ambivalence
  - → huge volumes of spam
- 2 Sending Method:
  - Botnets
  - → Low asymmetric bandwidth, widely distributed



# Transport-Level Characteristics of Spam

## Combining Observations: Low Penetration + Sending Methods

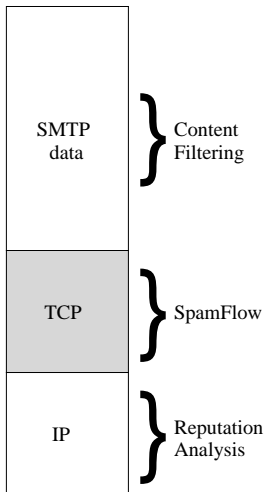
Volume + Methods + Economics → link/host resource contention



## Contention:

Contention manifests as TCP/IP loss, retransmission, reordering, jitter, flow control, etc.

# Understanding SpamFlow



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



# A Brief Diversion on TCP/IP

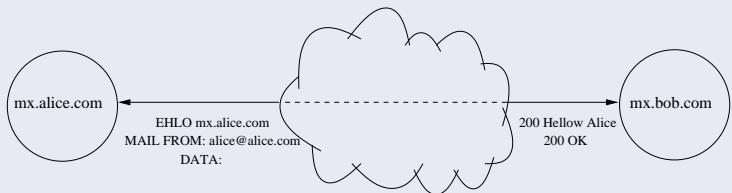
## Transmission Control Protocol (TCP):

- 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



# SMTP and TCP

## 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

How do Spam Connections Behave?

# How do Spam Connections Behave?

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

```

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

```





# How do Spam Connections Behave?

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

```

RcvQ  SndQ  Local                Foreign Addr          State
0      0      srv:25              92.47.129.89:49014    SYN_RECV
0      0      srv:25              ppp83-237-106-114 :29081                SYN_RECV
0      0      srv:25              88.200.2...
0      0      srv:25              92.47.12...
0      0      srv:25              ppp83-23...
0      0      srv:25              88.200.2...
0      0      srv:25              88.200.2...
0      0      srv:25              88.200.2...
0      0      srv:25              84.255.1...
0      25     srv:25              222.123...
0      28     srv:25              adsl-poo...
0      31     srv:25              222.123...
0      15     srv:25              222.123...
0      9      srv:25              88.245.3...
0      25     srv:25              78.184.1...
0      23     srv:25              190-48-3...
0      23     srv:25              dsl.dyna...
0      23     srv:25              ip-85-16...
0      23     srv:25              88.234.14...
0      23     srv:25              p5B0FBB5D.dip.t-d:11965  FIN_WAIT1
...

```

## 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

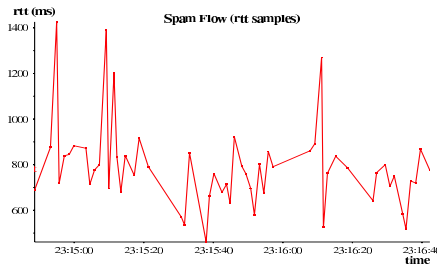
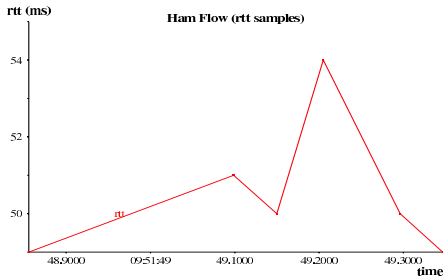


# What about RTT?

...building more intuition

Received: from vms044pub.verizon.net  
 From: "Dr. Beverly, MD" <b@ex.com>  
 Subject: thoughts  
 Dear Robert,  
 I hope you have had a great week!

Received: from unknown (59.9.86.75)  
 From: Erich Shoemaker <ried@ex.com>  
 Subject: Replica for you  
 A T4g Heuer w4tch is a luxury statement  
 on its own.  
 In Prestlge Repllcas, any T4g Heuer...



# Results

## CEAS 2008:

- “*Exploiting Transport-Level Characteristics of Spam*” [BS08]
- Offline analysis
- Utilize statistical machine learning methods
- Demonstrate  $> 90\%$  accuracy, precision, recall (w/o content or reputation!)
- Correctly identify  $\simeq 78\%$  of false negatives from content filtering alone
- See paper for details...



# Outline

- 1 Background
- 2 Detecting Bot-Generated Spam
- 3 Real-world Botnet Detection**
- 4 Current Research



# Obstacles to Deployment

## Obstacles to Deployment:

- Must be real-time
- Must be on-line
- Lots of “plumbing,” i.e. exposing transport-features to higher layers
- Training a supervised learner

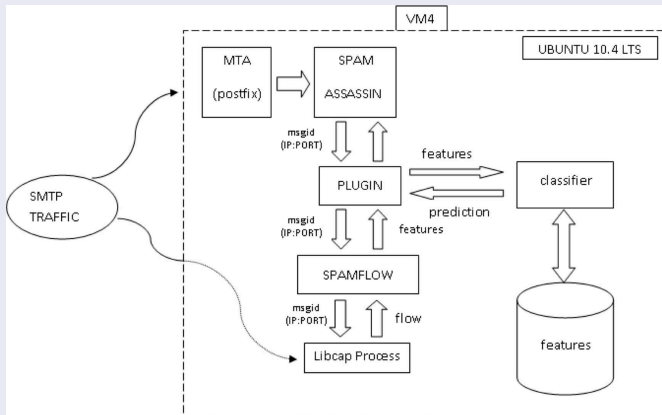
## USENIX LISA 2011:

- *“Auto-learning of SMTP TCP Transport-Layer Features for Spam and Abusive Message Detection”* [KBY11]
- Built a SpamFlow plugin for SpamAssassin
- Did the “hard” work



# SpamAssassin Plugin

## Plugin Architecture:



# Example Email

## Example Tagged Email:

```

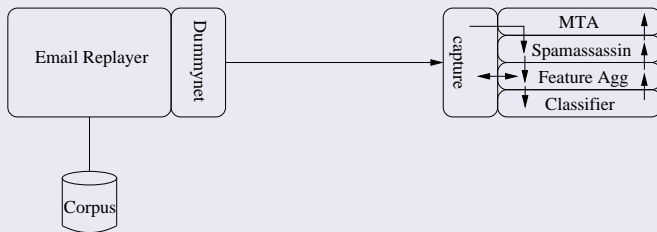
From Josephine@rsi.com Tue Feb 01 23:21:58 2011
Return-Path: <Josephine@rsi.com>
X-Spam-Checker-Version: SpamAssassin 3.3.1 (2010-03-16) on ralph.rbeverly.net
X-Spam-Level: **
X-Spam-Status: No, score=2.9 required=5.0 tests=BAYES_40,HTML_MESSAGE,SPAMFLOW,
UNPARSEABLE_RELAY autolearn=no version=3.3.1
X-Spam-Spamflow-Tag: 3792891725:37689,12,10,0,0,0,1,1,0,53248,34.464852,0.162818,
120.441156,148.297699,51.891697,5840,48,1,64
Received: (gmail 30920 invoked from network); 1 Feb 2011 23:21:57 -0000
Received: from cm-static-18-226.telekabel.ba (77.239.18.226:37689)
Received: from vdhvjcivivjbwbyhxnsfcvfwq (192.168.1.185) by bluebellgroup.com (77.239.18.226)
with Microsoft SMTP
Message-ID: <4D489025.504060@etisbew.com>
Date: Wed, 2 Feb 2011 00:20:48 +0100
From: Essie <Essie@hermes.com>
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.9.2.12)

```



# Lab Environment

## Lab Stress Testing:



- A “replayer” to emulate real-world load
- Utilizes a modified dummynet to emulate real-world network
- Reads a corpus (Enron, NIST TREC, etc)





# Auto-Learning

## Auto-Learning:

- Central problem in any supervised learner – how to train?
- We utilize the auto-learning functionality in SpamAssassin:
  - SpamAssassin returns a continuous *score* based on many, many tests
  - If other modalities (e.g. keywords, rule tests) indicate strong possibility of spam (high score) or ham (low score), use that as a *training example*
- Incrementally build the model
- Requires *no* human labeling or work!



# Production Experiments

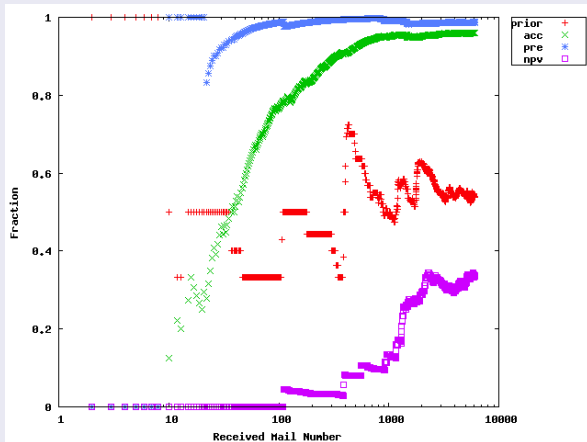
## January-March, 2011:

- Auto-learning thresholds based on spam distribution (normal,  $\mu = 16.3, \delta = 7.7$ )
- $\tau^+ = 16$  and  $\tau^- = 1$
- Yields training of 2,685/5,510 (48.7%) spam and 267/416 (64.2%) ham messages
- Experiments using Naive Bayes, C4.5 decision trees, SVM



# Auto-Learning Performance

## Auto-Learning Performance:



# Outline

- 1 Background
- 2 Detecting Bot-Generated Spam
- 3 Real-world Botnet Detection
- 4 Current Research**



# Current Research

## Lots of On-going Work:

- 1 Ph.D student, 1 graduating MS student, 2 current MS students
- Beginning work on 3yr NSF award (SDCI)



# Current Research

## Application to Other Domains:

- Attacks (automated) against web servers
- Can't rely on reputation and/or ports (as compared to SMTP spam)

## Detecting Botnet Hosting Infrastructure:

- Botnet CDNs – same requirements!
- Support scams (e.g. Canadian pharma)
- Provide mis/re-direction (Fast-Flux DNS, HTTP redir, proxying, etc)
- Capt Le Nolan to present next (from USENIX Security, 2011)



# Current Research

## Utilizing Transport Features:

- Adversarial learning to combat e.g. classifier poisoning
- Adversarial TCP/IP stack to cause suspected bot to perform *more* work, contributing to the feedback loop such that transport features are exacerbated
- Hardware deployment in NetFPGA, etc.

