

# SPAM BASICS



**Yue Duan**  
**Illinois Institute of Technology**

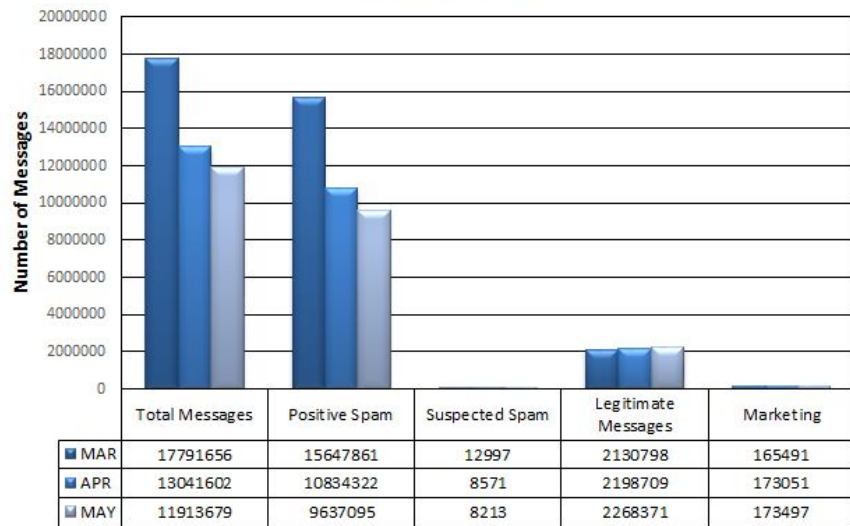
# SPAM

- A registered trademark
- Unwanted message was inspired by the 1970 SPAM Monty Python's Flying Circus.
- Was first applied to messages that were used to **scroll other messages off** user screens
- To respect the trademark, we call it “spam” rather than “SPAM.”



# SPAM

## 3rd QUATER SPAM STATISTICS FY2018-2019



# SPAM

## Common Email Spam File Types



**94%** of malware is delivered via email



Office doc files



Windows apps



Other

Source: Verizon report 2021

# SPAM

- Submit the same message to a **large group** of individuals.
- In an effort to force the message onto people who would otherwise choose not to receive this message.
- A message is spam only if it is both **Unsolicited and Bulk**.
  - Unsolicited Email is normal (examples: job enquiries)
  - Bulk Email is normal (examples: subscriber newsletters)

# HOW EMAIL WORKS

- The Simple Mail Transfer Protocol (**SMTP**) is an internet standard communication protocol for electronic mail transmission.

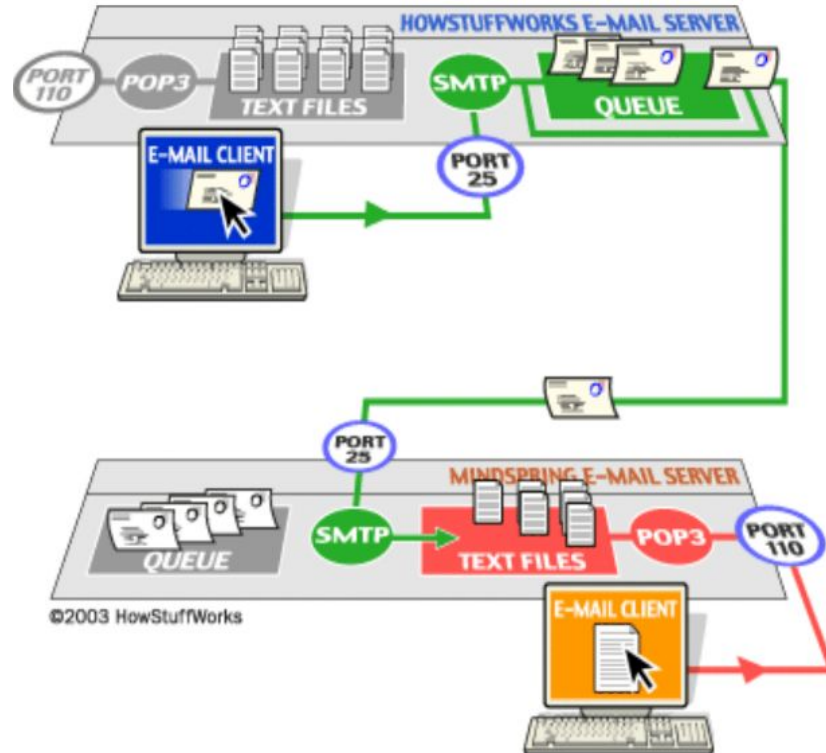
```
S: 220 smtp.example.com ESMTP Postfix
C: HELO relay.example.com
S: 250 smtp.example.com, I am glad to meet you
C: MAIL FROM:<bob@example.com>
S: 250 Ok
C: RCPT TO:<alice@example.com>
S: 250 Ok
C: RCPT TO:<theboss@example.com>
S: 250 Ok
C: DATA
S: 354 End data with <CR><LF>.<CR><LF>
C: From: "Bob Example" <bob@example.com>
C: To: Alice Example <alice@example.com>
C: Cc: theboss@example.com
C: Date: Tue, 15 Jan 2008 16:02:43 -0500
C: Subject: Test message
C:
C: Hello Alice.
C: This is a test message with 5 header fields and 4 lines in the message body.
C: Your friend,
C: Bob
C: .
S: 250 Ok: queued as 12345
C: QUIT
S: 221 Bye
{The server closes the connection}
```

# HOW EMAIL WORKS

- Sender (SMTP client) establishes a reliable communication channel
- Initiation
- Receiver and bounce address
- Data transmission
- Session ended

```
S: 220 smtp.example.com ESMTP Postfix
C: HELO relay.example.com
S: 250 smtp.example.com, I am glad to meet you
C: MAIL FROM:<bob@example.com>
S: 250 Ok
C: RCPT TO:<alice@example.com>
S: 250 Ok
C: RCPT TO:<theboss@example.com>
S: 250 Ok
C: DATA
S: 354 End data with <CR><LF>.<CR><LF>
C: From: "Bob Example" <bob@example.com>
C: To: Alice Example <alice@example.com>
C: Cc: theboss@example.com
C: Date: Tue, 15 Jan 2008 16:02:43 -0500
C: Subject: Test message
C:
C: Hello Alice.
C: This is a test message with 5 header fields and 4 lines in the message body.
C: Your friend,
C: Bob
C: .
S: 250 Ok: queued as 12345
C: QUIT
S: 221 Bye
{The server closes the connection}
```

# HOW EMAIL WORKS





# WHY SPAM STILL A PROBLEM?

- Email system design
  - Headers allow spoofing
- Identity concealing
  - Bot-networks
  - Open proxies
  - Open mail relays
  - Untraceable Internet connection
- Available bulk email tools

# EMAIL SYSTEM DESIGN

- SMTP designed for a trusting world
  - email is **not private**
  - can be **altered** en route
  - **no way to validate** the identify of the email source
    - **MAIL FROM** under total control of sender
    - Recipient's mail server only sees IP address of direct peer (recorded in the first **From** header)
- SMTP-AUTH (IETF RFC 4954)
  - an extension of the SMTP

# EMAIL SYSTEM DESIGN

- Headers are unreliable, can be used for spoofing
  - Insert fictitious email addresses in the From: lines

Check out the view



Inbox x

? Donald Trump <potus@whitehouse.gov>

to dft

-----  
from: **Donald Trump** <potus@whitehouse.gov>

to: dft@tweney.com

date: Wed, Oct 25, 2017 at 9:35 PM

subject: Check out the view  
-----

[MEMBER FOR \\$19/year.](#)

The view from my office is just amazing! It's really terrific!

```
<?php
$to      = 'nobody@example.com';
$subject = 'the subject';
$message = 'hello';
$headers = 'From: webmaster@example.com' . "\r\n";
mail($to, $subject, $message, $headers);
?>
```

# IDENTITY CONCEALING: BOT-NETWORKS

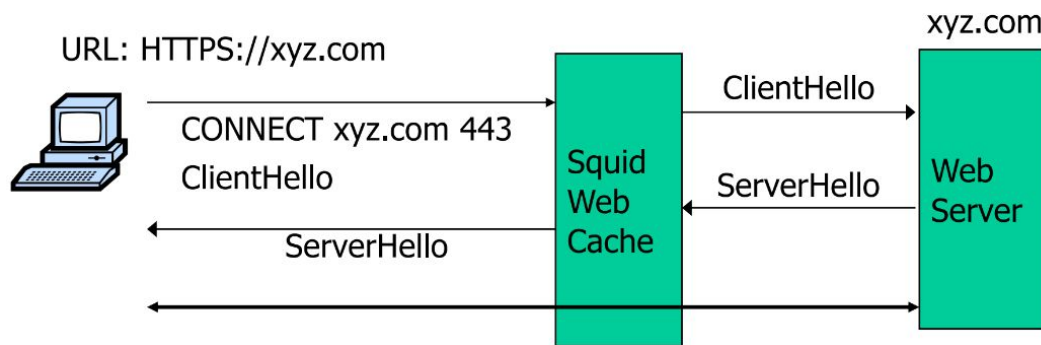
- Compromised machines running malicious software
- Once infected, spammer can send spam from it
- The bot software **hides itself and periodically checks for instructions** from the administrator
- Emails appear to come from **legitimate users**
- Example botnets:
  - Phatbot had 400,000 bots
  - Bobax : assimilated machines with high speed Internet connection

# IDENTITY CONCEALING: OPEN PROXIES

- An open proxy is one which will create connections for any client to any server, without authentication
  - SOCKS
  - HTTPS
- Possible for a computer to be running an open proxy server without knowledge of the computer's owner
- More difficult to detect when chain of open proxies used

# IDENTITY CONCEALING: OPEN PROXIES

- Web caching proxy (HTTP/HTTPS proxy) e.g., Squid

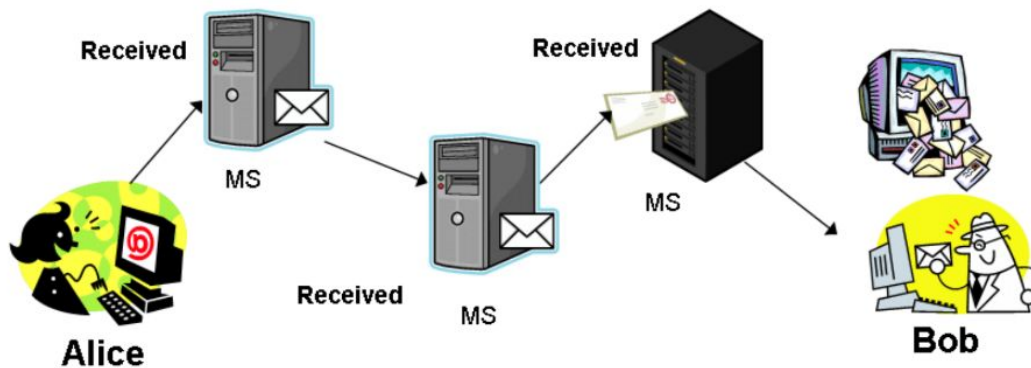


- To spam:

```
CONNECT SpamRecipient-IP 25
SMTP Commands
```

- Squid becomes a mail relay ...

# IDENTITY CONCEALING: OPEN MAIL RELAYS



- An email server configured to allow **anyone** on the Internet to relay email through it.
- Network address of spammer appears in one of the Received: headers
- Add **fake** Received: headers

# OPEN RELAYS VS. OPEN PROXIES

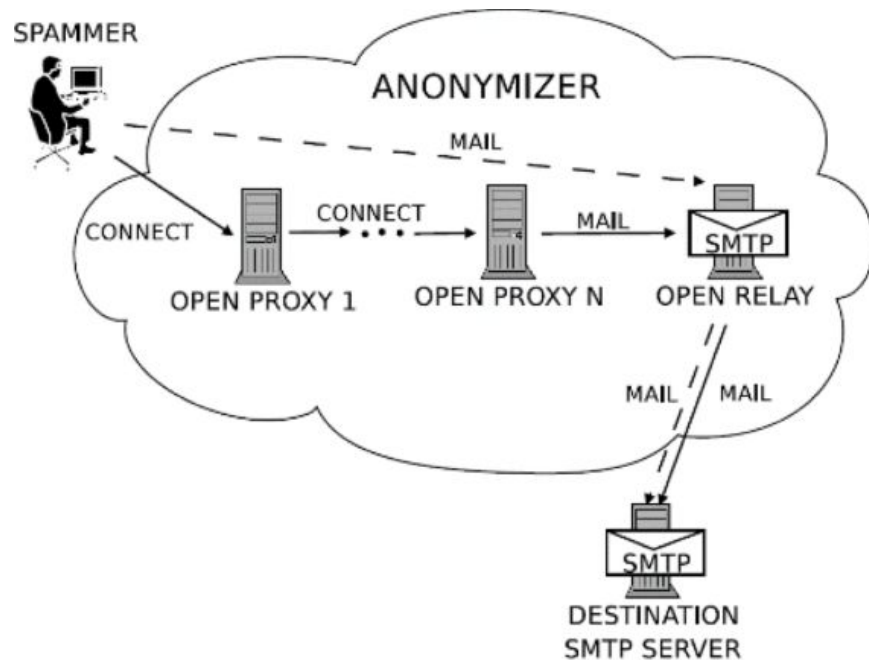
- HTTP proxy design problem:
  - Port 25 should have been blocked by default
    - Otherwise, violates principle of least privilege
- Relay vs. proxy
  - Relay takes list of address and send msg to all
  - Proxy: spammer must send msg body to each recipient through proxy.

==> zombies typically provide hacked mail relays



# COMBINING OPEN PROXY AND OPEN RELAY

- Establish TCP connection with Open Proxy1
- Connect with Open Proxy2
- Send email to Open Relay through this chain
- Forward to destination SMTP server

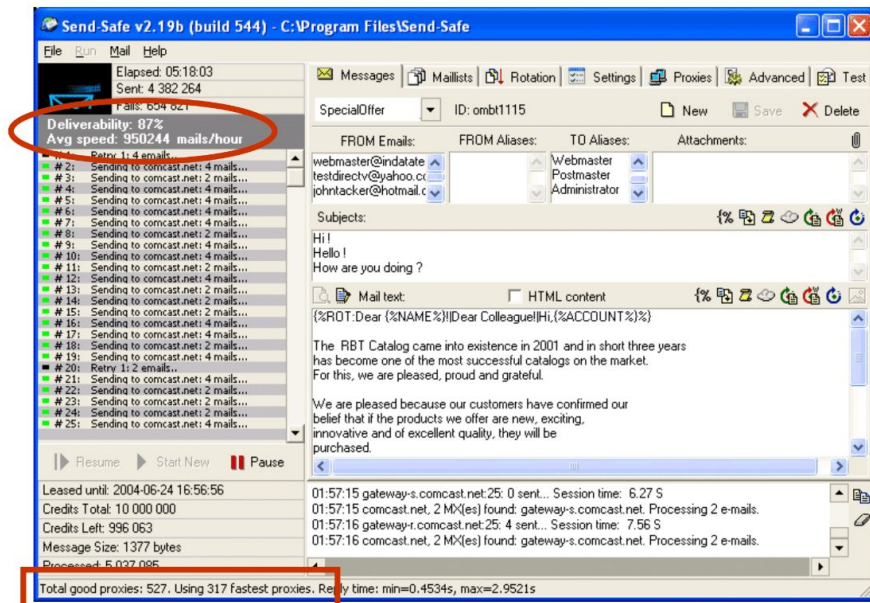


# IDENTITY CONCEALING: UNTRACEABLE INTERNET CONNECTION

- Public Internet cafes
- Free/stolen wireless connections
- Connections need not identify users
- Hence they need not hide network address
  - Send email directly to spam recipients
  - No way to associate email accounts with the spammer

# BULK EMAIL TOOLS ( SPAMWARE )

- SPAMWARE automates:
  - Message personalization
  - Mailing list and proxy list management



# SPAM COUNTERMEASURES

- Blacklists
- Graylists
- Puzzles and CAPTCHA
- Sender verification
- Spam filter

# SPAM COUNTERMEASURES: BLACKLISTS

- RBL: Realtime Blackhole Lists
  - Includes servers or ISPs that generate spams
  - e.g., spamhaus.org , spamcop.net
- Effectiveness (stats from spamhaus.org ):
  - RBL can stop about 15 - 25% of incoming spam at SMTP connection time
  - Over 90% of spam with message body URI checks
- Spammers evade blacklists by hiding its source IP address

# SPAM COUNTERMEASURES: GRAYLISTS

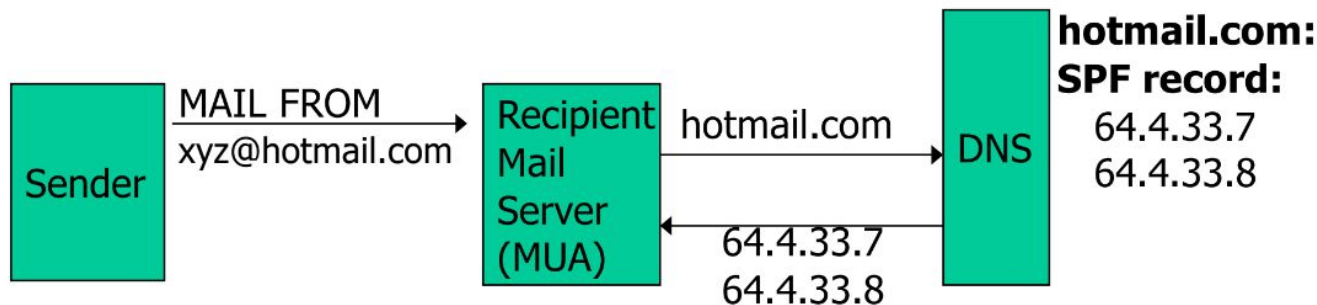
- Recipient's mail server records triples:
  - (sender email, recipient email, peer IP)
  - Mail server maintains DB of triples
- First time : triple not in DB
  - Mail server sends 421 reply: "I am busy"
  - Records triple in DB
- Second time (after 5 minutes): allow email to pass
- Triples kept for 3 days (configurable)
- Easy to defeat but currently works ok

# SPAM COUNTERMEASURES: PUZZLES AND CAPTCHA

- General DDoS defense techniques
- Puzzles : slow down spam server
  - Every email contains solution to puzzle where
  - challenge = (sender, recipient, time)
- CAPTCHA – Every email contains a token
  - Sender obtains tokens from a CAPTCHA server
    - Say: 100 tokens for solving a CAPTCHA
  - CAPTCHA server ensures tokens are not reused
- Either method is difficult to deploy

# SPAM COUNTERMEASURES: SENDER VERIFICATION I: SPF

- SPF: sender policy framework
- Goal: prevent spoof email claiming to be from Hotmail
  - Why? Bounce messages flood HotMail system



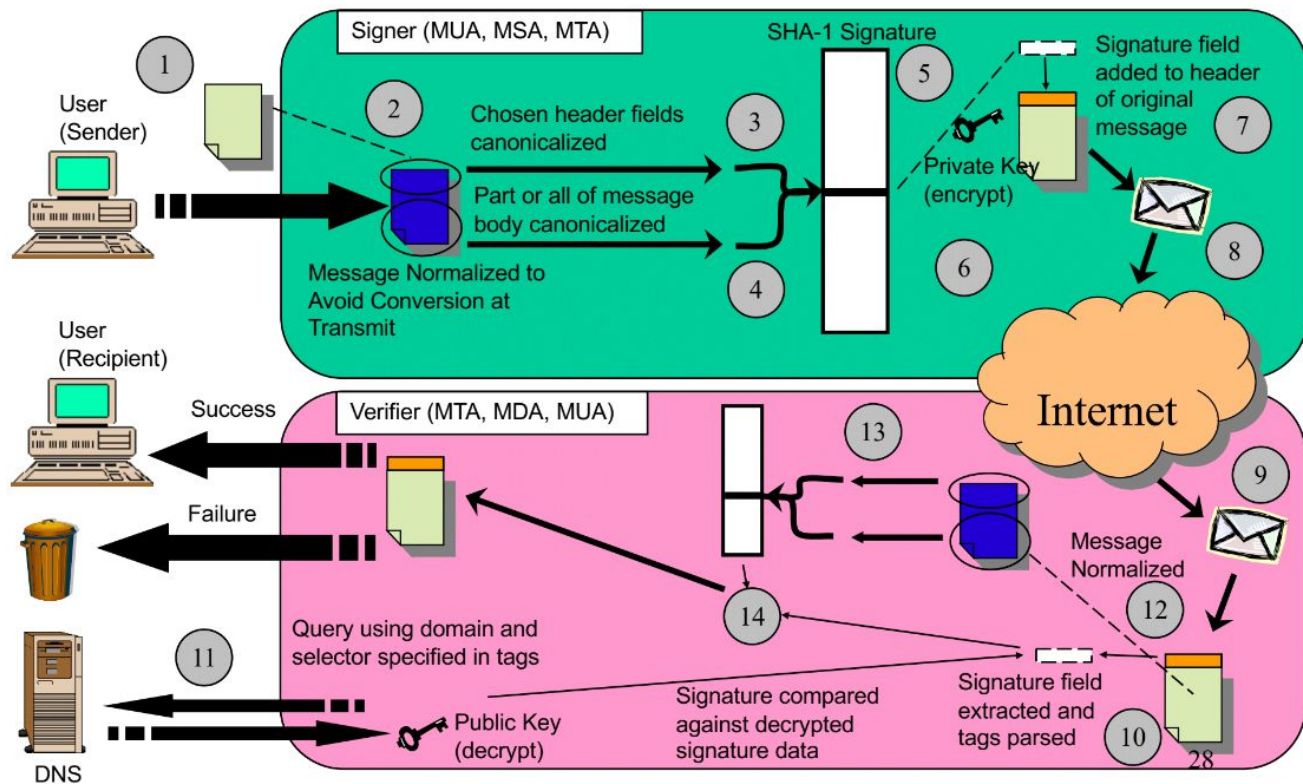
Check if SenderIP is in the list



# SPAM COUNTERMEASURES: SENDER VERIFICATION II: DKIM

- DomainKeys Identified Mail (DKIM)
  - Same goal as SPF
  - Harder to spoof
- Basic idea:
  - Sender's message transfer agent (MTA) signs email
    - Including body and selected header fields
  - Receiver's mail user agent (MUA) checks signature
    - Rejects email if invalid
  - Sender's public key managed by DNS
    - Subdomain: **\_domainkey.hotmail.com**

# SPAM COUNTERMEASURES: SENDER VERIFICATION II: DKIM



# SPAM COUNTERMEASURES: SENDER VERIFICATION II: DKIM

| Submitted to signer   | After signing  |
|---|--|
| <p>From: John Doe &lt;jdoe@anydomain.com&gt;<br/>To: Jane Doe &lt;jane@otherdomain.com&gt;<br/>Subject: Test message<br/>Date: Tue, 28 Feb 2006 18:00:20 -0700 (PDT)<br/>Message-ID: &lt;20060228010020.32345.5F7J@anydomain.com&gt;</p> <p>This is a test.</p> <p>Please reply to this message to confirm you have in fact received it.</p> <p>Thanks.</p> | <p><b>DKIM-Signature: a=rsa-sha1; s=newyork; d=anydomain.com; c=simple; q=dns; i=jdoe@anydomain.com; h=Received : From : To : Subject : Date : Message-ID; b=r6YHkli98DSewl2OPjIkd43SDeru78PI9iOu30wQdfE398KjHtGYJhBvCx65Mkl9</b></p> <p>Received: from mail.anydomain.com [10.2.3.4]<br/>by submitserver.anydomain.com with SUBMISSION;<br/>Tue, 28 Feb 2006 18:01:34 -0700 (PDT)</p> <p>From: John Doe &lt;jdoe@anydomain.com&gt;<br/>To: Jane Doe &lt;jane@otherdomain.com&gt;<br/>Subject: Test message<br/>Date: Tue, 28 Feb 2006 18:00:20 -0700 (PDT)<br/>Message-ID: &lt;20060228010020.32345.5F7J@anydomain.com&gt;</p> <p>This is a test.</p> <p>Please reply to this message to confirm you have in fact received it.</p> <p>Thanks.</p> |

Signature header added. Contains tags specifying various signing and key selection info (e.g. signing algorithm, signed header field info, key)

# SPAM COUNTERMEASURES: SPAM FILTERING

- Goal 1: No false positives
  - i.e., valid email ranked as spam
  - 0.02% more realistic
- Goal 2: No false negatives
  - i.e., spam email marked as valid
  - 8% more realistic
- Filters must be adaptive
  - Machine learning and data mining
  - Statistical methods
- Customizable policies per user group

# SPAM COUNTERMEASURES: SPAM FILTERING - CONTENT FILTERING

- Example: SpamAssassin
- a **rule-based** spam filter
  - Many rules to give scores for all fields in an email
    - e.g., Email header, special keywords in email, URLs in email
  - Final decision is the combined score compared with a threshold
  - Has false positive and false negative
    - False positive is very **damaging**! Nobody wants to lose an important email!
- Also contains Bayesian filtering to match a user's statistical profile
  - Need “ham” (wanted) and “spam” (unwanted) samples for training

# SPAM COUNTERMEASURES: SPAM FILTERING - CONTENT FILTERING

- Bayesian Spam Filtering
  - Tokenize mail content into words or phrases
  - Use a classified training set of spam and ham messages to derive conditional probabilities for each token
  - Use Bayes' Theorem to **calculate a probability** that a new message is spam based on these learned conditional probabilities
- Bayes' Theorem:  $P(A|B) P(B) = P(B|A)P(A)$

# SPAM COUNTERMEASURES: SPAM FILTERING - CONTENT FILTERING

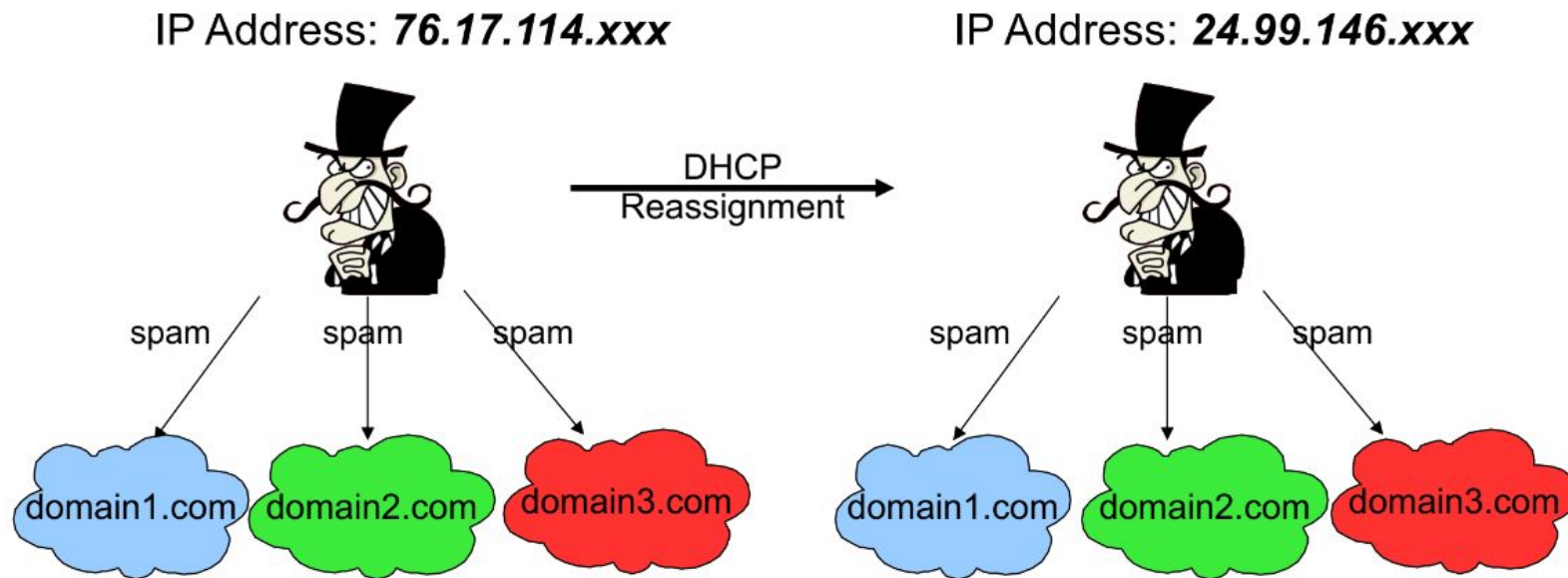
- Problems with Content Filtering
  - **Low cost to evasion**: Spammers can easily alter features of an email's content can be easily adjusted and changed
  - Customized emails are **easy to generate**: Content - based filters need fuzzy hashes over content, etc.
  - **High cost to filter maintainers**: Filters must be continually updated as content - changing techniques become more sophisticated

# SPAM COUNTERMEASURES: SPAM FILTERING - NETWORK-BASED

- Network-Based Filtering
  - Filter email based on **how** it is sent, in addition to simply **what** is sent
  - Network-level properties are less malleable
    - Set of target recipients
    - Hosting or upstream ISP (AS number)
    - Membership in a botnet (spammer, hosting infrastructure)
    - Network location of sender and receiver
- Key idea: find **invariants**

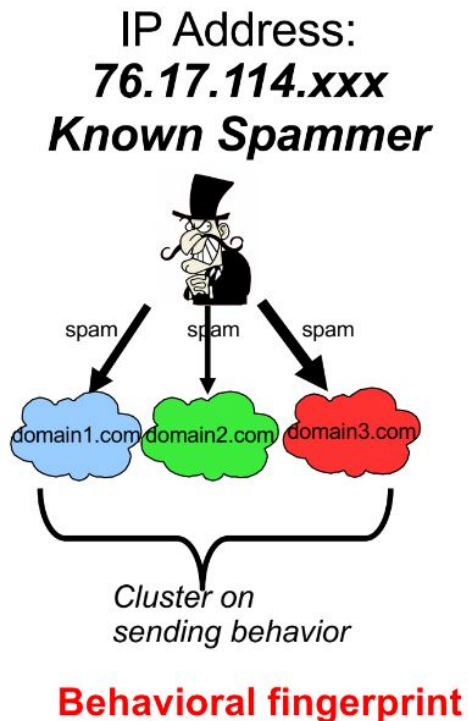


# SPAM COUNTERMEASURES: SPAM FILTERING - NETWORK-BASED



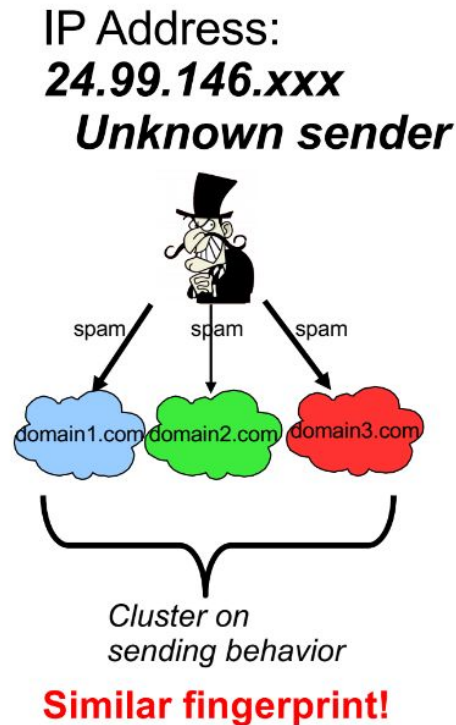
- **Spammer's sending pattern has not changed**

# SPAM COUNTERMEASURES: SPAM FILTERING - NETWORK-BASED



DHCP  
Reassignment

Infection



# SPAM COUNTERMEASURES: SPAM FILTERING - NETWORK-BASED

- Feature: **Distribution** of email sending volumes across recipient domains
- Clustering Approach
  - Build initial seed list of bad IP addresses
  - For each IP address, compute feature vector
    - volume per domain per time interval
  - Collapse into a single IP x domain matrix
  - Compute clusters
- Classifying IP Addresses
  - Given “new” IP address, build a **feature vector** based on its sending pattern across domains
  - Compute the similarity of this sending pattern to that of each known spam cluster

# SPAM COUNTERMEASURES: SPAM FILTERING - EVASION

- Problem: Malicious senders could add noise
  - Solution: Use smaller number of trusted domains
- Problem: Malicious senders could change sending behavior to emulate “normal” senders
  - Use additional features and combine for more robust classification
    - Temporal: interarrival times, diurnal patterns
    - Spatial: sending patterns of groups of senders

# SPAM BASICS

**THANK YOU!**  
**QUESTIONS?**