LABELLING SPAM THROUGH THE ANALYSIS OF PROTOCOL PATTERNS

Andrei Husanu & Alexandru Trifan
Bitdefender, Romania

Email {ahusanu, atrifan}@bitdefender.com

ABSTRACT

The increasing volume of unsolicited commercial email has driven the anti-spam industry towards taking advantage of every technique that can trim down the number of unwanted messages from the early stages to help reduce processing time and hence, system load.

This research paper aims to answer the question '*in how many ways can you send an email?'* Of course, in theory all SMTP servers behave according to the relevant RFC, but this is not necessarily the case in practice. Moreover, specific implementations behave differently, even if purportedly implementing the same set of rules.

The authors propose ways of fingerprinting the behaviour of various email-sending software. The actual contents of the messages are ignored. Instead, sending behaviour is analysed at the SMTP and TCP/IP protocol levels to find distinguishing patterns. Identification is based on several strategies including heuristics over packet sequence and length.

This is done with the purpose of identifying email messages originating from botnets and isolating them from those originating from various kinds of legitimate email servers. This is possible due to the way spam bots work, and their limitations.

INTRODUCTION

The first thing that comes to mind after reading the abstract is the highlighted question: in how many ways, really, can one send an email message? You just send it, and that's it. This is why it came as a surprise to us to discover that some spam waves define the TCP/IP protocol patterns associated with the source of the message particularly well.

The first signals began showing up when we started fine-tuning our custom SMTP servers, which are written in node.js. Selective logging of the TCP/IP packets revealed that, in some cases, packet sizes were standing out oddly.

A word about TCP/IP and MSS

SMTP is, of course, built on top of TCP/IP, a protocol which represents the foundation of reliable, ordered and error-checked delivery of data over networks.

In a nutshell, TCP accepts data from a stream, divides it into chunks and adds a TCP header to each chunk, creating TCP segments. TCP segments are then encapsulated into Internet Protocol (IP) datagrams and exchanged with peers.

Relevant in this context is a parameter of TCP called the maximum segment size (MSS). This represents the largest amount of payload data (not counting TCP headers), specified in bytes, that TCP is willing to receive in a single segment. For best performance, the MSS should be set small enough to avoid IP fragmentation but large enough to avoid overhead. The initial MSS is deduced by each side from the maximum transmission unit (MTU) size of the networks over which they communicate. It is then announced during the initialization of the TCP connection, in an attempt to optimize the performance of data transmission. Furthermore, TCP senders can use strategies that dynamically adjust the MSS according to the network conditions, all of these in order to minimize IP fragmentation.

The default TCP maximum segment size is 536 bytes, a value which bears close relation to the minimum network MTU size [1]:

```
MSS = MTU - sizeof( TCPHDR ) - sizeof( IPHDR );
```

The TCP and IP headers are, by design, variable in size. By convention, the default size for each of the headers is considered to be their minimum size (20 bytes each), hence the following:

```
MSS = MTU - 40;
```

There are several reference values for MTU sizes, depending on the transmission media. The following table is a non-exhaustive list of MTU sizes for different network media [2]:

Network	MTU (bytes)
16 Mbps Token Ring	17.914
4 Mbps Token Ring	4.464
FDDI	4352
Ethernet	1500
IEEE 802.3/802.2	1492
PPPoE (WAN Miniport)	1480

Table 1: MTU sizes for different network media.

Spam bots and why they are special

Spam botnets are the most common way of sending spam because they can easily be automated:

1. An outbreak of viruses and worms is sent in the wild, infecting systems and deploying a malicious application – the bot.
2. The bot registers with a command and control (C&C) server controlled by the botnet operator and is ready to receive and serve several commands, such as sending emails from via SMTP.
3. Spammers purchase email-sending services from the operator, providing them with templates and target address lists.
4. The operator instructs the compromised machines to send out spam messages.

One important aspect here is the great variety of systems that can be infected, among which we enumerate:

- Home-owned *Windows*-based PCs
- Email servers
- Servers that host unpatched *WordPress*/*Drupal* etc.
- Mobile phones
- Lately, even refrigerators and other devices belonging to the Internet of Things [3].

Bots belonging to the above categories have very different network environments, leaving traceable marks in the TCP/IP logs and creating distinguishable patterns.

THE SET-UP

In order to analyse the SMTP data and determine how it is split into TCP/IP segments, we created a simple SMTP server supporting the basic commands so we could simulate the receiving of an email message. The server was written in node.js to emphasize the different network events that occur as messages come in. We pointed some spam trap domains towards the server and we waited. Not for long.

In order to test how TCP/IP handles large chunks of data split over multiple write attempts, we created a small TCP/IP server hosted in the USA and a small client hosted in Romania that sends data to the server. Both have similar MTU sizes.

HOW TCP/IP SPLITS DATA IN MULTIPLE SEGMENTS

Using the simple client-server set-up described above, we started analysing how different data transmission scenarios end up on the server.

The server logs sessions of data it receives on each line with this format:

```
[<session #>] (<segment #>) <segment size> (<segment #>) <segment size> ...
```

For example, the following line:

```
[2] (1) 4320 (2) 800
```

Means that in session #2, the server received a 5KB buffer in two segments, one 4,320 bytes long, and one 800 bytes long.

Scenario #1

In this scenario, the client sends a single 5KB buffer with a single write call. The client code is:

```
client.write( buff5k );
```

Repeating this scenario at well-delimited intervals led to the following results on the server:

```
[1]  (1) 5120
[2]  (1) 4320    (2) 800
[3]  (1) 1440    (2) 1440    (3) 2240
[4]  (1) 1440    (2) 1440    (3) 1440    (4) 800
[5]  (1) 2880    (2) 1440    (3) 800
[6]  (1) 1440    (2) 1440    (3) 1440    (4) 800
[7]  (1) 1440    (2) 2880    (3) 800
[8]  (1) 1440    (2) 2880    (3) 800
[9]  (1) 1440    (2) 1440    (3) 2240
[10] (1) 1440    (2) 1440    (3) 1440    (4) 800
[11] (1) 1440    (2) 1440    (3) 2240
[12] (1) 1440    (2) 1440    (3) 1440    (4) 800
```

Observations:

- We notice the different ways the protocol splits the 5KB buffer into segments with sizes that are multiples of 1440, which is the base MSS.
- The only time the server receives segments with a size different from this pattern is when the remainder of the data needs to be transmitted.

Scenario #2

In this scenario, the client sends two chunks of 5KB each with two sequential write calls. The client code is:

```
client.write( buff5k );
client.write( buff5k );
```

Repeating this scenario at well-delimited intervals led to the results shown in Listing 1 on the server.

```
[1]  (1) 10240
[2]  (1) 1440    (2) 1440    (3) 2240    (4) 5120
[3]  (1) 1440    (2) 1440    (3) 1440    (4) 2240    (5) 1440    (6) 2240
[4]  (1) 1440    (2) 2880    (3) 800     (4) 2880    (5) 2240
[5]  (1) 1440    (2) 2880    (3) 5920
[6]  (1) 1440    (2) 1440    (3) 1440    (4) 5920
[7]  (1) 5120    (2) 4320    (3) 800
[8]  (1) 1440    (2) 8800
[9]  (1) 1440    (2) 1440    (3) 1440    (4) 800     (5) 1440    (6) 3680
[10] (1) 1440    (2) 1440    (3) 1440    (4) 5920
[11] (1) 1440    (2) 3680    (3) 1440    (4) 2880    (5) 800
[12] (1) 2880    (2) 1440    (3) 800     (4) 5120
```

Listing 1: Result of repeating Scenario #2 at well-delimited intervals.

```
[1]  (1) 5120  (2) 2880  (3) 1440  (4) 800
[2]  (1) 4320  (2) 800   (3) 1440  (4) 1440  (5) 1440  (6) 800
[3]  (1) 1440  (2) 1440  (3) 1440  (4) 800   (5) 1440  (6) 1440  (7) 1440  (8) 800
[4]  (1) 2880  (2) 1440  (3) 800   (4) 1440  (5) 1440  (6) 1440  (7) 800
[5]  (1) 1440  (2) 3680  (3) 1440  (4) 2880  (5) 800
[6]  (1) 4320  (2) 800   (3) 1440  (4) 2880  (5) 800
[7]  (1) 4320  (2) 800   (3) 1440  (4) 2880  (5) 800
[8]  (1) 1440  (2) 2880  (3) 800   (4) 1440  (5) 1440  (6) 1440  (7) 800
[9]  (1) 1440  (2) 1440  (3) 1440  (4) 800   (5) 1440  (6) 2880  (7) 800
[10] (1) 1440  (2) 1440  (3) 1440  (4) 800   (5) 1440  (6) 1440  (7) 1440  (8) 800
[11] (1) 1440  (2) 1440  (3) 2240  (4) 1440  (5) 1440  (6) 1440  (7) 800
[12] (1) 1440  (2) 1440  (3) 1440  (4) 800   (5) 1440  (6) 1440  (7) 2240
```

Listing 2: Result of repeating Scenario #3 at well-delimited intervals.

```
[1][1400] 'MIME-Version: 1.0\r\nX-Received: b[...]ues also were a factor.\r\nChrist,'
[2][1400] ' that are joined to the auditori[...] sight. In the 1830s the governm'
[3][1400] 'ent engineers determined that th[...]t/html; charset=utf-8" http-equi'
[4][1400] 'v="Content-Type" />\r\n</head>\r\n<b[...]ch at the absolute latest.\r\n\r\nTh'
[5][1400] 'e company makes indispensable so[...]d messages to any other processo'
[6][1400] 'r. Cognitive Behavioral Therapy,[...]to redeem his honour, thanks to '
[7][1400] 'Gizlof\'s schemes.<br />\r\nDaly wa[...]ts.<br />\r\n</body>\r\n</html>\r\n\r\n-'
[8][36]   '-e385b177227b0265de9baa986ec4--\r\n.\r\n'
```

Listing 3: Typical example with consistent segment size.

Observations:

- In the first session, both chunks of data are received in a single segment. Both write calls are processed by the link layer in a single transfer unit.
- In most other cases, the observations from Scenario #1 also apply here – the segment sizes are multiples of the base MSS and remainders.
- In some cases (1, 3, 5, 6, 8, 10), the two chunks of data are treated as a single 10KB block.
- In the other cases, the two chunks of data are transmitted separately (the segment sequence of the first 5KB chunk is highlighted).

Scenario #3

This time, the client also sends two chunks of 5KB each, but with two calls separated by a 100ms timeout. The client code is:

```
client.write( buff5k );
setTimeout( function( ) {
client.write( buff5k );
client.end( );
}, 100 );
```

Repeating this scenario at well-delimited intervals led to the results shown in Listing 2 on the server.

Notice that when the two write calls are separated by a 100ms timeout, the two chunks of data are split into two transports and split into segments accordingly (the segments belonging to the first transport are highlighted). On decreasing the timeout to smaller values, the probability of this behaviour drops from 100%, but it is still notable.

The above observation will prove to be very useful when analysing the data received by our small SMTP server listening in the wild.

SMTP SENDING PATTERNS

We started logging the network events in our small SMTP server, and after a while, some patterns started showing up throughout the huge log files.

The server logs events in the following format:

```
[<segment #>][<segment size>] 'received and possibly shortened data'
```

Official MTAs

Message transfer agents (or MTAs) usually consider data persistence the top priority. This is why the process of delivery often includes saving messages on the hard drive and sending the entire contents of a message to the network in a single write call. In a very congested environment, minimizing syscalls and letting the network layer do the splitting job may prove important.

This means that, at the other end, we should receive the message in a similar way to that in which we received data in Scenario #1: sequences of segments each with a size that is a multiple of the MSS. And this is, indeed, what we received.

Take the typical example shown in Listing 3. Notice that the segment size is very consistent and only the remainder of the message differs.

Of course, as the network conditions change, the MSS may improve to multiples of the base value, making the example shown in Listing 4 also very typical.

```
[1][1448] 'MIME-Version: 1.0\r\nX-Received: b[...]servation League.\r\nIn 1788 a cla'
[2][1448] 'ssical theater was built under h[...]January 2007 and returned to NAS'
[3][2896] 'A in an administrative position.[...]believe in me, and if you don\'t '
[4][2896] 'it\'s too bad. You will be sendin[...]lf to be a hacker at the time. G'
[5][138]  'uam to refuel, then hit the enem[...]57969aacd855bb3383b0b9a0f--\r\n.\r\n'
```

Listing 4: Another typical example – MSS improves to multiples of the base value.

```
[1][77]   'Received: from 192.168.1.3 (HELO[...]un, 23 Feb 2014 20:37:02 +0800\r\n'
[2][2896] 'Date: Sun, 23 Feb 2014 20:37:02 [...]IgaHJlZj0iaHR0cDovL2RldGFpbC50\r\n'
[3][329]  'bWFsbC5jb20vaXRlbS5odG0/c3BtPW[...]xMDU3NTI5ODY3\r\nLmpwZyIgLz4=\r\n.\r\n'
```

Listing 5: Relay servers can be configured in numerous ways, but this pattern stands out.

```
[1][520]  'Received: from PC20140213O9BEH[1[...]t\r\nContent-Disposition: inline\r\n'
[2][1440] '\r\n<!DOCTYPE HTML PUBLIC "-//W3C/[...]</P>\r\n<P style="MARGIN: 0cm 0cm '
[3][1440] '0pt" class=MsoNormal><SPAN style[...]NA <o:p></o:p></SPAN></P>\r\n<P st'
[4][1345] 'yle="MARGIN: 0cm 0cm 0pt" class=[...]al></SPAN></P></BODY></HTML>\r\n\r\n'
[5][5]    '\r\n.\r\n'
```

Listing 6: The headers are clearly sent in a distinct writing sequence.

```
[1][9366] 'Received: by 03e9fbce.2pt3t02.co[...]---=Part.739.2673.1393262439--\r\n'
[2][5]    '\r\n.\r\n'
```

Listing 7: The SMTP sequence for ending the data transmission is sent in its own segment.

```
[1][57]   'From: =?GB2312?B?uu635dTL?= <gk13qnadmin@it0668.com.cn>\r\n'
[2][2960] 'Subject: campfield294\r\nTo: "camp[...]AgICAgICAgICAgDQogICAgICAgICAg\r\n'
[3][111]  'ICAgICAgICAgICAgICAgICAgICC7tiDT[...]t_2rfkindysadvnqw3nerasdf--\r\n.\r\n'
```

Listing 8: Distinct FROM header.

SMTP relay servers

A slight variation of the above pattern is that seen in relay servers. Relay servers are SMTP servers that only forward the email, but not before adding their own Received header to the message. Of course, relay servers can be configured in numerous ways, but the pattern shown in Listing 5 stands out.

Notice that the received header is received in its own segment, clearly delimiting the two write sequences. The rest of the content is received just as if it were sent from a legitimate MTA (in one segment with the MSS a multiple of 1448, and another with the remainder of the data).

Clients that adjust the message contents on-the-fly

This category of clients clearly stands out because the behaviour highly resembles the email automation process of a bot generating a message from an email template.

Headers first, then entire body

In this pattern, the headers are clearly sent in a distinct writing sequence. Consider the example shown in Listing 6. Notice that the first segment contains only the message headers, while the body is sent in a distinct sequence of segments. It's as if the message headers were generated in a completely separate sequence – an indication that they were probably forged.

Also, we notice something that shows up so frequently we almost assigned it a sub-category of its own: the SMTP sequence for ending the data transmission is sent in its own segment, as shown in Listing 7.

Distinct FROM header

This pattern is a typical (not so smart) spam bot behaviour – just pick a random email address from a set, use it as the sender and then send the rest of the message contents.

The example shown in Listing 8 depicts this pattern. Notice that the FROM header is in its own segment, while the rest of the headers and message body are sent in standard size segments (with MSS values that are multiples of 1440).

Custom tailored headers

A small variation of the above pattern is one where more headers are sent in their own separate sequences, clearly indicating a spam bot behaviour.

Let's consider the example shown in Listing 9.

```
[1][399]  'MIME-Version: 1.0\r\nX-Received: b[...]RPYHYekEni@xxxxx.xxx>\r\nSubject:'
[2][29]   'Alert: Best Stock to Buy Now'
[3][184]  '\r\nFrom: Alberta Hendrix <stolber [...]a6d6aa2e90de7b468e09e64e2\r\n\r\n'
[4][848]  '--a643d6daa2e90de7b468e09e64e2\r\n[...]licopter. Picton is the only tow'
[5][1460] 'n in the Southern Hemisphere whe[...]e mountain, Dr. The Sri Lankan D'
[6][1460] 'rum Tradition is believed to go [...]w tip PRFC and you are willing t'
[7][1460] 'o hold a few weeks and see magic[...]not buying when I told you to do'
[8][1460] ' so.\r\n\r\nHappy Trading,\r\nI\'m Mike[...]the video cuts off the latter'
[9][1377] ' half of the piece. Included als[...]3d6daa2e90de7b468e09e64e2--\r\n.\r\n'
```

Listing 9: Custom tailored header.

```
[1][1400]  'MIME-Version: 1.0\r\nReceived: by [...]BlbmdbmVzIGJlY2FtZSBEdXJh\r\ndGVj'
[2][536]   'cyB0b28uIFRoZXNlIHRva2VucyBhcmUg[...]IHdpdGggQWppdGggaGF2aW5nIHRvIGNo'
[3][536]   'b29z\r\nZSBiZXR3ZWVuIGhpcyBsYWR5IG[...]QgYmVpbmcgcHVuY2hlZCBtb3ZlZCB0aH'
[4][536]   'JvdWdoIHRoZSBw\r\ndW5jaGluZyBtZWNo[...]YW4gZmlyZWQgc2V2ZXJhbCBzaG90cyB5'
[5][536]   'ZXN0ZXJkYXkuIEVsaXRlIElj\r\nZSBIb2[...]1pbHkuIEdvZCBidWlsdCBFdmUgYWZ0ZX'
[6][536]   'IgdGhlIGZhc2hpb24gb2YgYSBzdG9yZW[...]cyBidWlsZGluZywgYmVzdCBrbm93biBh'
[7][1400]  'cyB0aGUgSGVhbHRoIExvZGdlLCBpcyBs[...] of Popular Music. In March 2006'
[8][7000]  ' the book =\r\nwas released in mas[...]cZ5nZiBBHiZUNg/wCIFT+D\r\nObp1emv'
[9][1400]  'QO9dmK0ZWAQkgn5DnynViBk9oqz6RlVS[...]+WSTzLvZrM+gpZ2LMV5JOSZN9NVYy\r\ns'
[10][1400] 'y4KjAwSOPQ47iBpKQtShOKjlOTxAviIg[...]gZ4/wAplsZBpdVQ+PeHdtq/xMSfCR+34'
[11][1400] 'gde\r\nV131WAmu1HA7lWBxIalVbSWJc+1[...]q+pPE5N1laaLXJawFzM+Vb4m9OPTE6Vl'
[12][1400] 'Yu0jVn+NM\r\nftAnvXfs3DdjO3zxINqKF[...]UX3UdWsWszL02VyBnxNgjtiB0omG+y/3'
```

Listing 10: Fluctuations are characteristic of cellular networks, where MSS adjustments are very common.

```
[1][920]  'MIME-Version: 1.0\r\nX-Received: b[...]ng. There are several fountains '
[2][920]  'around the base of the tower and[...]he battle party. Garson obliged '
[3][920]  'with the performance heard on th[...]essage, Woodbey was in and out o'
[4][920]  'f jail for several years. A new [...]arried out to the same Euro NCAP'
[5][920]  ' regulations. Head, Laboratory o[...]than tripled within a short peri'
[6][920]  'od of time (feel free to check i[...]th or by the end of the 1st week'
[7][920]  ' of march at the absolute latest[...]ke Statler.\r\n</textarea>\r\n<br />'
[8][920]  '<br /><br />\r\n\t<br /><br /><br /[...]s chance for revenge against his'
[9][920]  ' brother. It was founded in 1957[...]azine\'s most popular sections. I'
[10][920] 't\'s a low resolution copy of a v[...]od of less than three years. Thi'
[11][583] 's is the second time the AHS Con[...]ef2b722e2923a8e43749350b9--\r\n.\r\n'
```

Listing 11: The TCP segments all feature a very small MSS value, characteristic of a very poor Internet connection.

Notice that the headers of the message are sent in three distinct sequences, the SUBJECT and the FROM header values clearly standing out.

Clients on cellular networks (mobile devices)

Another pattern that stands out is one where segments have highly fluctuating MSS values, ranging from the minimum value (associated with GPRS) to higher values (associated with 3G etc.) Such fluctuations are characteristic of cellular networks, where such MSS adjustments are very common (see Listing 10).

To double check, we started checking out the sender IP addresses, only to discover that most of them indeed belonged to cellular operators.

The sender IP address that generated the example shown in Listing 10 is from a class that belongs to a mobile operator in Peru. Spam messages coming from Turkey were also not uncommon.

Email servers with a poor Internet connection

This pattern could easily have been overlooked by our research, because the TCP segments are uniform in size but they all feature a very small MSS value, characteristic of a very poor Internet connection (see Listing 11).

This means that the sender is either a low-budget server with a questionable reputation (to say the least), or not actually a server, but a spam relay bot.

The '2048' fan club

The last pattern worth mentioning is a class of clients that send notably distinct sequences, each containing multiples of 1,024 bytes of data (2048 and 8192 being the most frequent). These

```
[1][1460] 'Return-Path: ddaztesa@lprsystem.[...]4pt">u</span>Pi6K<span style=3D"'
[2][588]  'color:#29594C; =\r\nfont-size:14pt[...]lor:#29594C; =\r\nfont-size:14pt">'
[3][1460] 'p</span>&OElig;599<span style=3D[...]w<span =\r\nstyle=3D"color:#29594C'
[4][588]  '; font-size:14pt"> </span>&yuml;[...]"color:#29594C; font-size:14pt">'
[5][1460] 'x</span>r&aelig;P&Phi;<span =\r\ns[...]3D"color:#29594C; =\r\nfont-size:1'
[6][588]  '4pt">I</span>&#218;c3&zeta;<span[...]or:#29594C; =\r\nfont-size:14pt">o'
[7][1460] '</span>&ldquo;H&#224;&#230;<span[...];<span =\r\nstyle=3D"color:#29594C'
[8][588]  '; =\r\nfont-size:14pt">s</span>&#1[...]pan =\r\nstyle=3D"color:#29594C; ='
```

Listing 12: The sequences are split into the same segment patterns.

```
[1][1388] 'MIME-Version: 1.0\r\nX-Received: b[...]hile, Zoano PharaohMan is captur'
[2][72]   'ing NetNavis and transforming th[...]followers of Gregar, including Z'
[3][1388] 'oano SparkMan. Nino Valenti, and[...]an being kidnapped. He married l'
[4][1388] 'ocal woman Marie Longley, and be[...]f you\'re tired of playing the ma'
[5][1388] 'rket for mediocre gains then you[...]hone. This is absolutely revolut'
[6][1388] 'ionary and as we get closer to c[...]hn Croan et al. Meanwhile, Zoano'
[7][1388] ' PharaohMan is capturing NetNavi[...] to finish him off.<br />On Nove'
[8][991]  'mber 3, 2007, a man named Indran[...]4107d645756d0cb785590ad45--\r\n.\r\n'
```

Listing 13: Apparent noise coming from variations in the MSS value.

sequences are, most of the time, split into the same segment patterns, as can be seen in the example shown in Listing 12.

At first glance, we thought this fluctuation might be attributed to network fragmentation. However, high fragmentation usually triggers a mitigation reaction consisting of a rapid MSS adaptation. If segment #2 had been affected by fragmentation, the MSS for all subsequent segments would certainly have dropped from 1460 to a lower value. This is proof that our assumption is correct and that there is indeed a '2048' fan club.

CONCLUSION – WHY IS STUDYING THIS BEHAVIOUR USEFUL?

This new approach to analysing messages comes with some clear advantages.

The ability to isolate a group of patterns, associate them with a type of sender and then check for them in a live environment can contribute (in some cases, decisively) to the sender IP reputation. Also, when analysing a very consistent spam flow, different patterns can aid the isolation of different and sometimes new types of botnets and of the networks they work in. Of course, the most notable examples are botnets consisting of mobile devices.

Drawbacks of this method

While trying to extract patterns, we often stumbled upon apparent noise coming from variations in the MSS value due to network fragmentation. Let's look at the example shown in Listing 13, which is not uncommon.

Note that segment #2 is obviously shorter and might mistakenly be labelled as coming from a separate sending sequence. What actually happened was that segments #1 and #2 were initially a single segment with an MSS of 1,460 bytes, which proved too much for the network conditions at that time. The MSS was immediately adjusted to compensate for this and all future segments shared this common value.

The above example shows that extracting relevant TCP connection patterns requires in-depth strategies that help overcome apparent noise.

Improving existing work

While researching existing work in this field, we found the most notable contribution to be from *Postfix*, with the *Postscreen* daemon. It serves as the first layer of a multi-layer defence against spam by checking the conformity of SMTP clients with the protocol. Being a rule-based system, it can definitely be improved by adding TCP-connection-level rules based on the above patterns that weigh in on the decision.

REFERENCES

[1] Postel, J. RFC 879 – The TCP Maximum Segment Size and Related ... 1983. http://tools.ietf.org/html/rfc879.

[2] The default MTU sizes for different network topologies. 2004. http://support.microsoft.com/kb/314496.

[3] Proofpoint Uncovers Internet of Things (IoT) Cyberattack. 2014. http://www.proofpoint.com/about-us/press-releases/01162014.php.

[4] Postfix Postscreen Howto. 2010. ttp://www.postfix.org/POSTSCREEN_README.html.