



HOW DOES ANTI-SPAM WORK?

BY PIERLUIGI STELLA
CTO AT NETWORK BOX USA, INC.

NETWORKBOXUSA.COM

C O N T E N T S

07 HOW SPAM GETS THROUGH

04 HOW ANTI-SPAM WORKS

07 MAIL TRANSFER AGENTS



ANTI-SPOOF



ATTACHMENTS



ZIP FILES

" THIS EMAIL IS SO CLEARLY SPAM, HOW COULD YOUR SYSTEM MISS IT? "

I often speak to clients about spam that makes it through our filters. A typical question being, *"This email is so clearly spam, how could your system miss it?"*. More than likely, the email contains words such as *"male enhancement"*. Or the infamous *"Viagra"*. Worse yet, sometimes the word *"sex"*. It would, naturally, seem obvious that the software should spot such words, and block any emails containing them.

By all means, it isn't something we always discourage. However, consider the following possibilities:-

1. I'm emailing a friend, saying I'll be going to Essex
2. Or I'm purchasing a unisex t-shirt
3. Or I'm a doctor, writing to a patient about his ED, using words like the ones mentioned above from a purely professional standpoint

These are real life situations where blocking words result in the blocking of legitimate emails, and many of them, for that matter.

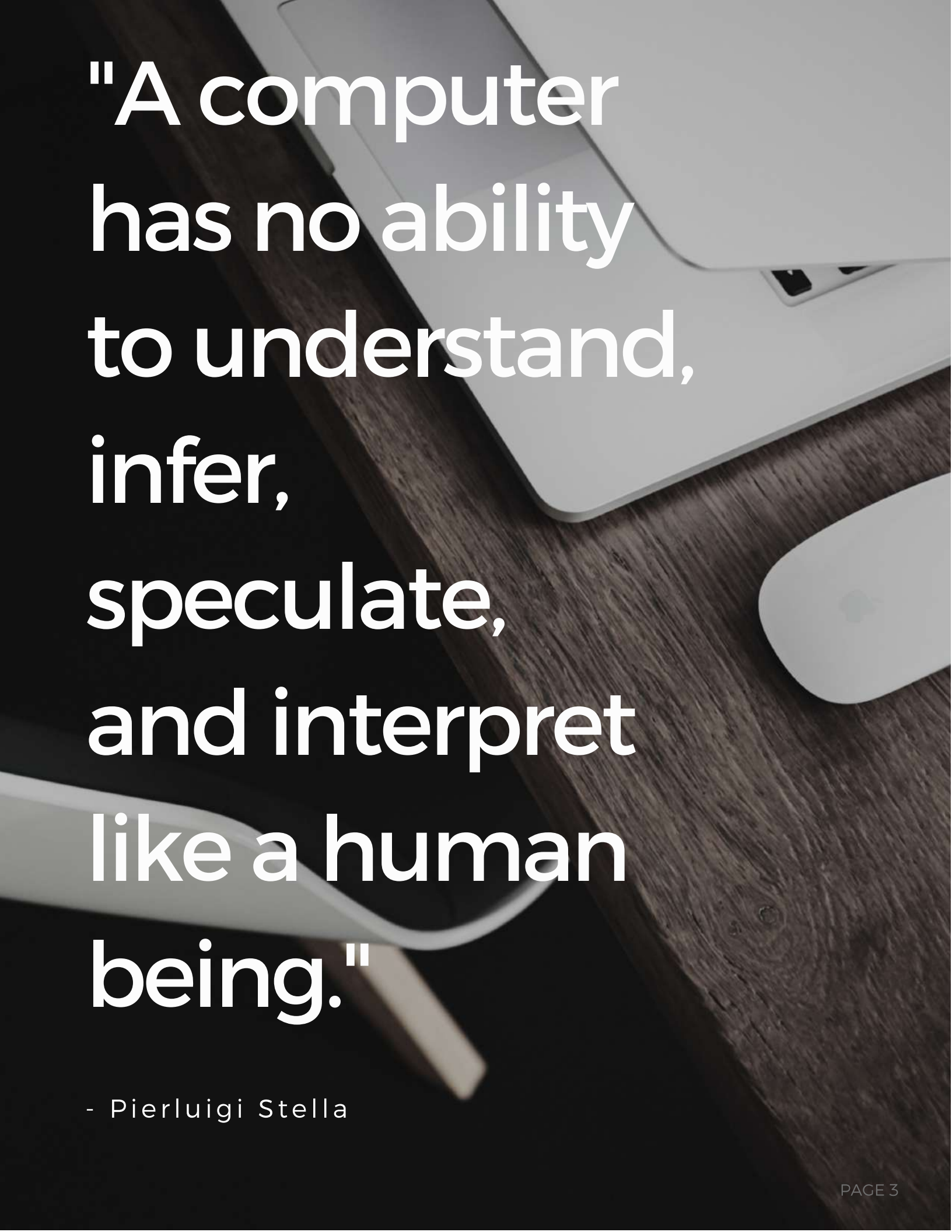
True, for a business, blocking emails because of the words they contain isn't unheard of but we must be very careful in what we choose to block. For instance, blocking the word that starts with F and ends with K is a good idea. I can't think of any legitimate use of this word, unless I'm trying to get myself fired, and that word is part of a nasty email I'm sending my boss. And I'm very sure we can all come up with a lengthy list of such words we do not want to see in an email, especially in an office setting.

HOW DOES ANTI-SPAM WORK?



" WE CANNOT BLOCK AN EMAIL BECAUSE IT CONTAINS TEXT THAT APPEARS IMMEDIATELY OFFENSIVE. "

At home, things are a little different. Applying such filters to personal emails may prove limiting to certain individuals who are, simply put, prone to profanities. But at the workplace, yes, we can definitely be more restrictive. However, I must reiterate, such restrictions must be applied very thoughtfully. We cannot block an email because it contains text that, to our intelligence, appears immediately offensive.



"A computer
has no ability
to understand,
infer,
speculate,
and interpret
like a human
being."

- Pierluigi Stella

HOW DOES ANTI-SPAM WORK?

A computer has no such '*understanding*'. Even machine-learning and AI, which are increasingly popular, have a hard time making such decisions; and those are based on very large super computers and lots of training.

A small device scanning your emails simply can't apply machine-learning. Yet. So, when you look at an email and think "*how can this possibly get through*", consider it from the viewpoint of a computer – the poor thing has no ability to understand, infer, speculate, interpolate, and interpret the way human beings do. It also doesn't have the malice to think beyond the actual words, to the intended underlying meaning of the sentence. If it sees "*male enhancement*", it's merely two words juxtaposed, nothing more.

At this point you might ask, "*how then does Anti-spam work*"? The answer isn't simple at all.

The first line of defense is provided by RBLs (*realtime blackhole lists*), databases listing the reputation of URLs and IP addresses. The most well-known of these is likely Spamhaus. We check many such lists to see if the sender's server IP address or domain is listed as a known source of spam or malware. If that's the case, we accept the SMTP connection solely for the purpose of retrieving the sender email, and then logging it for reporting purposes.

Had there been no need to log and report, we wouldn't even bother accepting those connections. They're coming from an IP address that's listed as compromised in reputable databases. Querying such databases isn't free for businesses such as ours. An individual can run a single query, but the hundreds of thousands of queries we run every hour, require a subscription.

Once the email passes this test, we check the SPF record (<http://bit.ly/2meY8Ex>). Although an SPF record isn't mandatory, if it is set, it must be respected. If the check fails, the email is blocked as a spoof.

HOW DOES ANTI-SPAM WORK?

Another type of DNS query we run is against the Name Server itself of the domain sending the email. When you receive an email, typically you want to reply. A legitimate sender will always have a complete set of DNS records, namely A, MX, NS and so forth. If the domain sending the email has no such records, you can be sure it isn't legitimate.

So, why accept that email to begin with?

We then check the recipient email address against our clients' users database. Spammers use a technique called Directory Harvest Attack (DHA) to send thousands, if not hundreds of thousands, of emails to a specific domain, literally making up possible email addresses, with the hope that john.smith@yourdomain.com exists. By checking against your database, we obtain two advantages. First, if the recipient doesn't exist, we don't accept the email. Why bother accepting and scanning an email if there's no one to send it to? Second, we blacklist the sender's IP address for a period of time. This is done because often, spammers executing DHA send many emails per minute to the same domain. Sometimes it's an actual barrage. By blacklisting the IP, we avoid accepting such emails, and we reduce network clutter and CPU usage. But we also achieve another objective. If by chance the spammer does manage to correctly guess one email address, having blacklisted their IP, we don't risk accepting and allowing an email that shouldn't be allowed in the first place. And yes, there are definitely failsafe measures in place for accidental misspellings of email addresses by legitimate senders.

" THE ENVELOPE OF AN EMAIL IS PART OF THE COMMUNICATION BETWEEN THE TWO SERVERS. "

The techniques listed above are a partial example of what we call 'envelope scanning'. The envelope of an email is part of the communication between the two servers, namely the one sending as well as the one receiving the email, and it's a part of the process the end-user never sees.

Once the email is received, the server discards the envelope. A bit like the butler opening your letters for you then discarding the envelopes.

At this point we start the body scanning. The body of the email is what you see in your inbox, and it contains headers, body text, and attachments.

The headers can be practically anything. Your mail client (*Outlook for most users*) recognizes the headers and only shows you a few such as sender (*which we call header:from*), recipient (*you*), subject, timestamp. As a general rule, you won't see anything else. However, there are other headers.

For example, one that's relevant to you is the reply-to header. If the reply-to differs from the "header:from", when you click "reply", the email will be addressed to the reply-to.

Other headers identify all the servers which handled that email. An email generally goes from the sender's workstation (*client*) to the sender's email server. From here, it goes to your server, and then your email client will download it on your workstation. However, this journey can be a lot more complicated.

HOW DOES ANTI-SPAM WORK?

There can be several of what we call Mail Transfer Agents (MTA) in between. For instance, the Network Box itself is an MTA. It intercepts incoming email for scanning purposes, before delivering it to your server. The sender might have such a scanning device as well, which scans for private or confidential information, or for viruses outbound. Should confidential information be found, the sender could use an encryption system, which is yet another MTA. You see how quickly the chain can extend? Each MTA leaves its trace in the headers of the email, adding two lines at least per, one which says "received from <name> and <IP> of the server sending the email", and another stating name and IP of the MTA itself.

“ SPAMMERS SPOOF SENDERS’ EMAIL ADDRESSES, TO MAKE YOU BELIEVE THE EMAIL IS COMING FROM SOMEONE YOU KNOW. ”

Other such headers will be related to what actions the MTA took on that particular email. If the MTA is an anti-spam/antivirus, it may say “*scanned by...*” and show the results of those scans. Some MTAs insert signatures. One such signature that is becoming increasingly accepted is the DKIM (or its newer version called DMARC). It’s a form of authentication of the sender’s server, one of many ways the industry is trying to address the pernicious problem of email spoofing. The reason why I’ve written so much about headers is because we do scan those headers in many ways. There is, of course, a lot we can learn about the history of that email by analyzing those headers.

I recently wrote a paper about anti-spoof. Spammers spoof senders’ email addresses, to make you believe the email is coming from someone you know; often your own CEO, for example. By analyzing the headers, we can tell when the email sender was spoofed, and we can block those emails, without even scanning them for viruses or spam.

Another way headers are useful is remember how I mentioned every MTA handling an email leaves its own name and IP in a header?

Who’s to say it didn’t attach something malicious to a perfectly legitimate email? In such cases, we run RBLs again, but this time on IP addresses and domain names we find in the headers.

If one of the MTAs is found to be blacklisted, the email is promptly blocked. We cannot take the risk of allowing through an email handled by a compromised server.

And you, as a user, shouldn’t want that email either. Even if that email stands to bring a potential \$50,000 project to your company, is it worth the danger of having a new form of ransomware delivered straight to your computer? At the expense of your entire network? Do you really want to risk it? While it is, at the end of the day, your choice, meaning the email can always be released, accepting emails from compromised servers entails a potential liability we strongly believe you should not accept.

HOW DOES ANTI-SPAM WORK?

Finally, we come to the body of the email. This will contain plain text, the same text repeated inside an HTML tag with html code, and any attachments.

The body text is what most users think of as the part that reveals whether something is spam or not. For computers though, that is likely the most useless part from which to make such determinations. We humans arrive at those conclusion by using skills computers don't have. The computer will analyze the bytes and words and, as we've seen earlier in this paper, that is truly not a good way to determine whether an email is spam or not.

Having said all that, one part of the body that can prove useful are URL links within the body of the email. Spammers often use links, disguising them as something else, to lure the user into clicking. Once you click, you end up on a rogue server, which downloads a Trojan to your computer. Analyzing these URLs is therefore very important. However, even this is a useless chase at times. Setting up a new domain is easy, takes a matter of minutes. Hackers set them up, use them in an email campaign that lasts several hours, then abandon them.

" HACKERS KNOW HOW OUR INDUSTRY BEHAVES, THEY KNOW THEY HAVE A LIMITED AMOUNT OF TIME TO MAKE THE BEST OF NEW URLS. "

Find your phone,
keys, anything

By the time the industry gets around to blacklisting the URL or the corresponding IP, it's already too late. We see this happen over and over. Hackers know how our industry behaves, they know they have a limited amount of time to make the best of new URLs so, they get them ready, use them briefly, then move on to new ones. The sheer volume generated in those few hours may gain them whatever it is they're trying to achieve, and makes it worth all the work involved in the initial set up.

[Finally, we come to the attachments.](#)

At this point, it's clear that emails often travel through several servers, each doing something with it. As the information gets transmitted over the internet, it also passes through many, many routers and other types of devices.

Nowadays, the industry has pretty much standardized how bit codes are represented. When networks were first invented though, there were many proprietary standards. Two that many of us may be familiar with are ASCII and EBCDIC (*a proprietary character code by IBM*). The major difference between these is that EBCDIC uses an inverted bit encoding. Without going too much into specifics, if the machines exchanging information are each using one of these two character sets, what starts as an A on the ASCII machine could end up as something completely different on the EBCDIC machine. Imagine the conflicting end results, and this issue is further compounded by the representation of binary data in the computers' registries.

" WHY AM I MENTIONING ALL THIS NOW? "

Because attachments often contain binary data. Take for example JPG images, they're binaries. Any attachment, short of a notepad txt file, will contain some form of binary. Sending such data across the Internet 'as-is' and hoping it will arrive at its intended destination, uncorrupted, is beyond wishful thinking.

The solution to this is binary-to-text encoding. The original encoding mechanism was called UUEncode (*Unix to Unix*). Today, we use several different kinds. Base64 being the most common, MIME is another; and there are many others. When viewing an email, your email client knows the languages of encoding and translates the encoded text/binary into something your computer can understand. However, when we scan an email in transit, we need to deal with encoded characters so the first thing we do is '*decode*' it, as in extract the actual binary in a form the Network Box understands.

" THE FIRST THING WE DO IS APPLY A POLICY ON THE TYPE OF FILE. "

Now we have a file in a universal format – could be PDF, DOC, ZIP, etc.

ZIP deserves special mention. Zip compressed files are called archived files, something most Windows users are familiar with. What many don't realize, however, is there are over 700 different types of archiving mechanisms. Take Linux as an example. We use something called tar, and we compress with something else called gzip.

Without opening/unzipping a compressed file, we cannot scan it properly, potentially allowing a virus to get through. That's why being able to open each of the 700+ types of archives is critical. Once we've done that, again, we have a file in the format the final user would see, and can now analyze it for threats and spam.

So the first thing we do is apply a policy on the type of file.

We can determine if the file is, for instance, a Windows executable file, even if the file is disguised as something else. Changing the file name and extension may 'trick' your Windows computer, but it should not confuse a good email scanner, which ought to recognize the true file type and true mime type of a file, determine if it's possibly executable in any way, and block it if it is.

Another possible policy is to simply look at the file extension and just block any extensions we don't want. Network Box currently lists over 40 such extensions. Any email containing an attachment file with any such extension is instantly properly blocked without further analysis.

"A COMPUTER CANNOT MAKE 'INTELLIGENT' DETERMINATIONS."

To overcome the issue of Zero-Day, at Network Box, we have one final arrow in our quiver, and that's Z-Scan. I won't go into detail what Z-Scan is right now but for more information, the reader can check this link (<http://bit.ly/2n7KCCK>).

One brief conclusion after this long dissertation.

Deciding if an email is spam or a threat is not as simple for a computer as it might appear to be for a human. A computer cannot make '*intelligent*' determinations. AI requires far too much computing power to be available for such tasks today. Therefore, we need to resort to down-to-earth options, such as determining if the source of the email is reliable, if it is who it claims to be, and all the many other mechanisms I've explained thus far. And to conclude, we still resort to signatures when necessary; but truly (*and only*) as a last resort.

I hope you've found this paper useful and informative.

HOW DOES ANTI-SPAM WORK?

FACEBOOK /NetworkBoxUSA

TWITTER /networkboxusa

LINKEDIN /NetworkBoxUSA

YOUTUBE /NetworkBoxUSA

PINTEREST /NetworkBoxUSA

BLOG blog.networkboxusa.com

WEBSITE networkboxusa.com

ADDRESSES

2825 Wilcrest
Suite 259
Houston, TX 77042

832.242.5757
info@networkboxusa.com