Internal Release Only

TECHNICAL MEMORANDUM

To: General Distribution

Project: General Information

From: David M. Ihnat

CC:

Date: 15 May 08 Last Modified: 05 Jun 08

Subject: E-Mail Overview

---

1. INTRODUCTION

This memorandum was created to explain and discuss the E-Mail systems and mail delivery and processing at a high level. It is intended for non-technical users.

2. OVERVIEW

E-Mail delivery is far more complex than the mail client (e.g., Outlook) or web mail interface you may use to receive and send your mail, and much of this complexity is outside of local control. Once a message leaves your mail server, it's at the mercy of intervening relaying mail servers, Internet Service Provider networking policies, and recipient mail servers, clients, spam filters, etc.

This complexity, combined with actively malicious activity—spam and *malware* (viruses, Trojans, etc.)—acts to create a system that doesn't always respond in the manner we expect or wish.

In addition to the external matters beyond our control, actions on our part can affect the stability and reliability of our mail services. Limits and controls can be placed on behavior of mail users, but as with most security measures, these can and will result in restrictions on some forms of utility. On the other hand, foregoing these restrictions and relying on restrained behavior by mail system users can, when such restraint hasn't been carefully exercised, result—and has resulted—in inadvertent interruption or degradation of mail service.

In following sections of this memorandum, we will discuss Internet E-Mail in general, and issues that affect it.

3. INTERNET E-MAIL

Internet Electronic Mail (E-Mail) is a complex and constantly evolving interaction between innumerable computers, managed by unrelated entities that rarely communicate directly with one another. Instead, compliance with a number of documented standards and conventional procedures is intended to permit interchange of messages across the Internet.

Unfortunately, at the time the mechanisms underlying the E-Mail infrastructure were designed, there were no such things as crackers or spam. Consequently, layers of programs and filters intended to protect against these corrupting influences have both complicated and, at times,

---

adversely affected the reliability of E-Mail delivery. This is an unavoidable consequence of the necessity to deal with such perversions of the basic system.

3.1 MAIL SENDING AND DELIVERY

In concept, the process of sending and receiving electronic mail is simple. A message is created—originally only text, today such a message can have complex and active embedded formatting, actual programs, and attachments. This message is handed to a program that "knows" how to examine the recipient address and, from that, determine how to contact another program over the Internet on the recipient mail server and hand the message over to it for final delivery to the intended recipient.

Of course, there are many subtleties between your keyboard and the recipient's. First, let's give you some common definitions used in the industry; you've heard most of them before, but may not have known exactly what was meant by the terms. We'll try to explain them in "English", meaning that more technical readers may have an urge to point out shortcomings; please don't bother. We're trying to keep it simple (and yet, because it *is* a complex system, it's still going to be complex to describe.)

- *Internet Message Access Protocol (**IMAP**)*. An alternative method to POP for accessing messages from a MTA, allowing the messages to remain on the MTA computer instead of being downloaded.
- *Mail Client*. This term refers to a program acting as a MUA.
- *Mail Server*. This is a computer running a MTA.
- *Mail Transfer Agent (**MTA**)*. The program that actually delivers or receives electronic mail, either from within a local organization or network, or to/from the "outside world".
- *Mail User Agent (**MUA**)*. A program website that manages the user-related tasks of accepting delivered mail, preparing mail to be sent, and usually managing the storage and organization of sent and received messages.
- *Post Office Protocol version 3 (**POP3**)*. This is a defined standard for MUAs to request messages from a MTA for download, usually into a Mail Client.
- *Simple Mail Transport Protocol (**SMTP**)*. This is a defined standard for MTAs to exchange information to carry out the task of recognizing, authenticating and accepting or sending mail messages.

3.1.1 Mail Transfer Agent (MTA)

This is a program that has two fundamental tasks. First, it accepts connections from programs (*Mail User Agents*, or **MUA**s) on a local network in order to collect messages to be delivered to either local users or users outside the local network. It handles the tasks of validating the message formats, authenticating senders, and saving messages if necessary until they can be delivered to a recipient MTA.

Note that your message may not be delivered directly to the recipient's MTA. It may go through several intermediate machines, for various reasons—security, or the recipient may not run their own mail server but rather rely on their Internet Service Provider (ISP) do actually handle mail receipt and delivery.

Also, there's not just one program everyone uses as an MTA. As long as it "speaks" a common "language" (the SMTP protocol), any program can be an MTA. There are a number of common programs, both commercial and free, that are used in this role; some examples are Microsoft Exchange, Sendmail. Postfix, and Exim, just to name a few.

3.1.2 Mail User Agent (MUA)

The MTA speaks a "language" that's incomprehensible to most people, and not at all convenient to use for those who *do* understand it. Instead, the MUA fills the role of "speaker to humans". It's essentially a program that carries out the necessary task of collecting a message to be sent, acquiring messages from the proper MTA that are destined for its user, and usually provides common mail support functions, such as address books, managing and storing sent and received messages, and displaying messages.

No matter what other functions the MUA carries out, its primary purpose is to "talk to" one or more MTAs to actually send and receive mail. Most users will be familiar with the requirement to define the "Incoming server (POP3 or IMAP)" and "Outgoing server (SMTP)" when setting up their mail accounts; that's the link between the MUA and the MTA.

3.1.2.1 Mail Clients

This term is commonly used to indicate a MUA that's a program actually installed on the end-user's computer. Common mail clients are *Microsoft Outlook*, *Mozilla Thunderbird*, or *Qualcomm Eudora*. Many of these programs tend to incorporate functions that aren't strictly related to electronic mail, such as schedule/calendars.

3.1.2.2 Web Mail

This is a common method of accessing electronic mail without having to load a Mail Client onto your computer. Instead, the software that acts as the MUA resides on a web server, and is accessed as a web page. The software on the web server carries out the normal functions of "talking to" the mail server, and usually stores sent and received messages on the web server.

The main advantage of using web mail is that you can access it from any internet-connected computer. The main disadvantages are that your mail is on the host computer—you may have to pay storage fees, and many are uncomfortable leaving private communications in the hands of whoever is running the web site. Also, most web mail interfaces don't offer as many additional functions as local mail client programs, and can be very slow, depending on the network connection.

3.1.3 Mail Sending and Delivery Wrap-up

Ok, so just to make sure things are clear; the process for sending a message is:

1. A Mail User Agent—either a Mail Client such as Outlook, or a Web Mail interface such as Outlook Web Access—is used to edit a message, create attachments, and do any formatting, and to build a list of recipients.
2. Once the message is submitted ("Send"), the MUA contacts the configured Mail Transfer Agent, authenticates the user (usually through a stored login and password), and hands the message off to be delivered.

3. The MTA may, if it's lucky, be able to talk directly to the MTA handling the recipient's E-Mail. More often, it needs to hand it off to an intermediate MTA—and this can happen multiple times—but eventually the message should reach the final recipient's MTA.
4. The recipient's MTA will hold the message for delivery until either a Mail Client or a Web Mail client connects with it to request entire messages and any attachments be transferred to it (POP3), or just the message headers (IMAP) be transferred.
5. Once the user requests to view a message, the MUA formats and displays the message, and allows such operations as deletion, forwarding, archiving, etc.

An important thing to notice in this cycle—the message is actually out of your and your mail system's control shortly after it is sent. As soon as your MTA can get rid of it, it's at the mercy of all those intermediate mail servers and, of course, the recipient's mail server and Mail Client.

3.2 FLIES IN THE OINTMENT

All of the above is how the original authors of the electronic mail systems and standards expected everything to work. All mail servers are trustworthy, and only people who really want to talk to each other will send mail. Alas, it's not that way, and wasn't for long even back then.

We have two major problems that have to be addressed on a day-to-day basis—*crackers*, and *spammers*. Today, they're almost inextricably entwined.

3.2.1 Crackers

Crackers[1] are crooks. They're trying to use either flaws in your computer's operating system, mail software, web browser, or network security to attempt to corrupt your system and network. By corrupt, we can mean everything from just play a prank, through data destruction, to taking over your computer to capture private information and send it off for identity theft, or to take over your machine to turn it into a "spambot", a machine to send out spam.

Often, in the earliest days, this was the relatively innocent one-upmanship that is practiced most often by adolescent males. Unfortunately, today this is big money—spam, identity theft, etc.

The most common means for gaining control of your system are through *social engineering*, or *malware*.

Social engineering is simple—they get you to do their work for them. Click on this web site URL to see something or get something…and when you do, you activate a program with your rights on your computer. Usually it also exploits some flaw in your browser to allow it to install software. Social engineering can also be that E-Mail that promises you've won, say, a prize—if you'll only fill out a form and send it back (usually asking for your SSN, or other private information.)

Malware is a shorthand term that's come into use for *malicious software*. Viruses, Trojans, and Worms are the main culprits.[2] These almost always come into your computer through E-Mail;

---

[1] The term you'll often hear used in the press for this particular crime is *hacker*. That's offensive to a lot of us old-timers, because it meant something totally different. Back In The Day, a hacker was a particularly skillful computer programmer or hardware developer; one who invented 'hacks' that accomplished tasks that the manufacturers often said couldn't be solved. Thus, the preferred term by us is *cracker*, because they're trying to crack into your system or network. And they're criminals, by out lights.

[2] Broadly speaking, a *virus* is a program that exploits some programming error in your computer's operating system to automatically install itself; it usually has to get started by some normal system activity, such as

visiting an infected web site; inserting an infected CD or other removable media in your computer; or activating an infected E-Mail attachment.

3.2.1.1 Defenses

There are three lines of defense against these infestations. First, firewalls—special software or hardware appliances designed to examine all traffic between a local network and the Internet, watching for unauthorized attempts to exploit services on the local network—is the most common "border" defense.

Secondly, *anti-virus* programs monitor a system, attempting the almost impossible task of trying to keep up with the constantly varying methods these programs use to try and slip in. They often scan E-Mail messages, all files that are copied or imported, and try to watch website content. Individuals will usually have a single copy for each computer; networks will usually have a centrally managed version.

The final line is a combination of various *inoculators* that attempt to prevent the system from getting infected (e.g., SpywareBlaster) and *anti-spyware scanners* that look for a large body of malware that attempts to steal information, or to take over your browser to pop up advertisements. These differ from anti-virus programs usually in the depth of their defense—they're targeted, usually, at a specific set of known offending programs, and don't do the extra coverage of monitoring E-Mail, websites, etc.

The problem with the latter two approaches are that they're *reactive*—they have to see an example of a piece of malware to figure out what it looks like, so they can recognize it in the future. But, at the moment, these are the best we have.

3.2.2 Spam

Unfortunately, pretty soon after the Internet was available outside academia and research, someone got the idea to send a message to every address he could find[1] to try and make a sale. From that point, it's been a continually escalating war between the spammers and users.

Spammers are always attempting to take over computers to send out their spam, and trying to disguise it so it can slip by defenses; users are always trying to trap it before it can flood their mailboxes and systems. To this end, they make use of malware as described earlier, usually to carry two tasks.

First, they want your computer. If they can get their software on it, it can be taken over to send spam for them. And this software is sophisticated—networks of literally 10,000 machines are "sold" to spammers by the developers of this software. These machines take orders from controllers—other computers (also usually *owned*, or taken over, by the spammer's software), accepting spam for redistribution.

Secondly, they want good addresses. A known good address means they can send a message to it and expect it most likely to be seen by a human—*if* they can get past the defenses in place.

---

clicking on an infected website, inserting an infected CD or floppy, etc. A *Trojan* needs the user to take some activity to start it—such as clicking on an innocent-seeming E-mail attachment. A *worm* is specifically a virus that knows how to replicate itself over a network.

[1] A DEC salesman named Gary Thurek is commonly "credited" as the first spammer, 30 years ago—May 3rd, 1978—although it wouldn't be called spam for another 15 years.

Some people ask why spammers DO this? Can it be that profitable, really? How many people really order "that little blue pill", or HGH hormone? The answer lies in how very *many* messages they send out. If they can take over a *botnet* of 10,000 controlled computers, it's no problem to send out, say 5,000,000 spam messages around the world. If only 0.5%--that's ½ of 1 per cent—actually order something, that's 25,000 orders. In that many people, they're going to find that many suckers. And all that E-Mail is free, thanks to their pirating of machines to turn into zombies sending that spam.

This is Big Money. It's infested by organized crime organizations, and funds research groups every bit as skilled as any the industry can field to fight back against them.

3.2.2.1 Defenses

Defense against spam consists of a combination of user education and software.

3.2.2.1.1 User Education

Computer system users need to learn what practices are risky, and avoid them. Receiving E-Mail from an unexpected source, asking the user to click on an embedded website address, or to open an attachment, should always be treated with suspicion—and *don't click that link! Don't open that attachment!* Sending addresses can be forged; if you receive an attachment from someone who you think you know, but aren't expecting to send you such an item, check with them before opening it.

Avoid sending mass mailings to large numbers of unverified addresses from your own mail server. In any group of 2000-3000 random addresses, it's almost a certainty that hundreds will be infected with spam programs that will harvest your address and pass it on to receive spam.

If something strange pops up on your computer—usually asking you to click on something to "protect" your system—*don't do it*. If you can, leave it and get someone knowledgeable to examine your system. If you don't have someone, run an anti-virus scan.

Avoid risky websites. Virtually anything in China (.cn), Korea (.kr); gambling sites, and, of course, sites of "questionable content".

3.2.2.1.2 Mail Filters

The foremost method to block spam today is some form of filter. This usually takes one or more of four forms:

- Filtering at the Internet Service Provider, if they provide your mail server.
- Contracting for a third-party security mail service to carry out more extensive filtering.
- Running a mail filter on your own mail server, if you host your own mail.
- Running Mail Client filters—for instance, Outlook has an internal Junk Mail filter; and third-party filters such as PopFILE can be installed on certain configurations of mail clients.

In all of these cases, the filters in question are complex, and use a wide range of techniques, all with the intention of examining incoming (and sometimes outgoing) mail to determine if it's spam or "real". This is a difficult task; the tools, usually combined into the filtering program and managed by it, are usually used in combination, and include but aren't limited to:

- *Bayesian filters*. Named after English mathematician Thomas Byes, this refers to a type of filtering that doesn't just look for specific words or phrases, but rather examines characteristics of the entire message to attempt to classify it as spam or real. Moreover, it "learns" the characteristics of messages on the fly; the more messages it examines, the better it gets at identifying spam. Moreover, as spam changes, it picks up these changes and incorporates them into its identification. This by far is the most common core of spam identification filters today.
- *Real-Time Blackhole Lists (RBL), Spam Lookup Services (SLS)*. These are usually websites that can be queried by mail servers to determine if a message has been sent from sites known to send spam; the result can be used to either mark the message as questionable, or reject it. The good ones are quite useful; however, as there are a large number available to choose from, and each uses different criteria to identify a sending address as a spammer, care must be taken when using this technique.
- *Blacklists and Whitelists*. Lists of addresses explicitly compiled to unconditionally accept or reject mail from given domains or users.
- *Block by Country*. If the given site never expects messages from, say, Russia (.ru), China (.cn), or other foreign countries, they can be blocked. This is of less utility than some other approaches, since the spammers can forge their sending domain, but it does catch a certain number of spam messages, and spam defense is a matter of degrees of protection.
- *Greylisting*. This is a relatively new technique that simply refuses acceptance of an inbound message from a site it's never seen before for a small, random amount of time. It depends on the fact that real mail servers will always retry in a few minutes; spammers program their sending programs to try to blast as many messages as possible, and won't retry.

There are double-fistful of other, more specific techniques, but you get the idea—use as many characteristics and helpers as possible to make the determination of "spam" vs. "mail".

Unfortunately, these techniques require a great deal of "tuning", and the interplay can be complex; when implementing a spam filtering solution, there's a period of time where this is visible to end users. And there is always the possibility of falsely identifying real E-Mail as spam—this is called a *false positive*. Short of accepting all spam and letting the user sort it out, there's no way to avoid this. Fortunately, it's possible, through whitelists and other techniques, to force known good senders to be accepted. Users of a mail system should be told about the anti-spam solutions, and should explicitly test E-Mail to and from important correspondents to make sure they make it through the filters.

4. CONCLUSION

The end result: There is no good solution to spam or crackers; all we can do is try to identify them and keep them off our systems. It's an ongoing and vicious war, with both sides continually trying to outwit the other. Fortunately, we do have tools at our disposal to fight this war, and in the end, the majority of mail does work.