# Sending Email from a Residential ISP

Vedika Bang
*Georgia Institute of Technology*

Brian Teachout
*Georgia Institute of Technology*

## Abstract

This paper is an anecdotal experimental exploration of emailing from a home internet connection. We investigate restrictions enforced by Internet service providers on residential customers. We find that email delivery is possible from a residential network, so we also attempt to assess spam filtering of eight different email servers based on DNS-driven email authentication. Analysis of these results suggests email authentication is not heavily relied upon by email providers.

**Keywords:** Email, Spam, DNS

## 1 Introduction

Every day, 122 billion unsolicited or abusive email messages, also called spam, are sent worldwide. That accounts for about 85% of global email traffic. [20] But wait, is it SPAM or a mistakenly tagged message because some of the terms in your message are detected in (trained) algorithms employed by spam filters? There is a lengthy thread on Reddit where people frequently complain about their emails not being delivered. Have you ever waited for an email from your alumni organization only to discover it two months later in your mail portal's "junk box"? Yes, SPAM is irritating; nevertheless, being blocklisted or tagged as spam despite being a legitimate sender is unfortunate. Is blackholing or hellbanning [5] required simply because the computational power to process every single email costs a fortune? Is SPAM ruining email? Nobody, especially mail providers, likes SPAM! Spam filters created by various email providers may take all or none of the characteristics into account - providers may block email addresses based on the patterns they detect; nevertheless, this raises the likelihood of false positives, which surely has a bad impact on the user experience.

In our Network Security and Measurement course, we read Spamalytics: a paper describing the Storm botnet, how it was used to send spam, and some estimations for conversion economics [15]. During the class discussion, some were skeptical as to whether a similar event could occur now. This led to the question: Is it even possible to send email from a residential Internet service provider (ISP) connection to a free webmail provider and have it delivered to their inbox? These ubiquitous email services were chosen as "Email is now an oligopoly, a service gate kept by a few big companies which do not follow the principles of net neutrality." [7]

This paper is a record of our work on that question. We explore the restrictions imposed by residential ISPs and then evaluate some of the filtering policies used by common free webmail providers. We do this by sending emails with a range of email domain authentication configurations, from compliant with fully implemented DMARC to in violation of all authentication parameters. In section 2 we cover a brief history of email and spam, introduce some relevant email terms, then discuss the tools we used for email authentication with the Domain Name System (DNS). In sections 3 and 4 we provide an overview of our efforts on evaluating ISP connections and configuring our testing setup. Then in section 5 we present our results and observations. Finally, in section 6 we discuss some of our limitations as well as ways the project can be expanded and improved.

## 2 Background and Related Work

### 2.1 Email and Spam

Spam (v.) The word's (not so) linguistic origins can be found in a Monty Python's Flying Circus comedy, where it was used as part of a group's chant, "SPAM, SPAM, SPAM, Lovely SPAM!" until they were told to stop. This chant later evolved into the term for unwanted, unwelcome, and unsolicited digital communication that is sent out in large quantities. [11]

Spam's history is entwined with the development of email. Gary Thurek, an American marketer pitching his company's computer products over ARPANET, sent the first email to roughly 600 of the 2,600 recipients in 1978. Thuerk asserted that his email resulted in an additional $12 million in sales. However, many of the recipients of his email became quite irate and complained to the US Defense Department, which managed ARPANET. [6] Over time, spam has undergone a significant change. Spam has been used for a variety of purposes. Making a clear distinction between "Solicited Marketing Emails" and "Unsolicited Marketing Emails" is crucial.

The strategies used by adversaries are forever evolving. Spam in the 1990s and 2022 are on opposing extremes of the spectrum. The 1990s were the "Wild West" of email marketing because there were no rules, sender verification, or ISP limits in place. ISPs had to take action to reduce spam and irrelevant emails once the detrimental effects of "Just send it" became obvious and email abuse escalated. Early in the postmaster era, in the late 1990s and early 2000s, ISPs began to use primitive filtering to find unwanted and unsolicited emails. In response, spam filters have begun to gain traction in the market. However, the filtering was not precise enough, which resulted in numerous false positives. ISPs would generally use filters like header filters, content filters, blocklists, permission filters, and rule-based filters to detect spam. [8]

The SenderScore was the next development in the mid-to-late 2000s toward keeping email safe and free of spam. By including sender authentication measures like SPF and SenderID, as well as following them with DomainKeys Identified Mail (DKIM), false positives were reduced significantly. The advancement of machine learning and big data nowadays can greatly assist ISPs in reducing spam.

## 2.2 Email Operational Overview

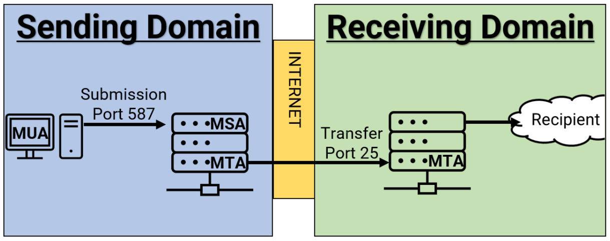


Figure 1: A Simplified Diagram of Email Submission

The basic structure of email is important to our test setup. A simplified view of email-sending architecture is seen in figure 1. The Message User Agent (MUA), on the left side of the graphic, is the email client application that, in the general case, originates an email. Common MUAs include the Microsoft Outlook desktop application, Mozilla Thunderbird, and webmail clients. The MUA sends this email message to the Message Submission Agent (MSA), which is responsible for receiving email messages for routing from clients of the domain. This communication primarily occurs using TCP port 587, which requires client authentication, although TCP port 465 is sometimes used as an alternative [19]. The MSA then passes the message to the Message Transfer Agent (MTA). The MTA initiates an SMTP exchange with a remote server MTA over TCP port 25 [17]. The remote domain then works to deliver the message to the appropriate recipient. The delivery mechanisms are not immediately relevant to our work in this paper. While port 25 was originally used for submission and transfer, it is increasingly common to conduct unauthenticated transfers over that port, while authenticated submission uses port 587 instead.

## 2.3 Email Authentication with DNS

DNS is a vital component of digital communication. DNS converts domain names to IP addresses and is ubiquitous on the Internet. When a user enters a website into the browser, a DNS lookup begins—working in the background, referencing DNS servers located worldwide. Without DNS, the mail server would not know where to deliver the mail because an email address must be mapped to an IP address. Email addresses always include the domain name (userID@domainname), which explains why email would not function properly without DNS. However, there are other record types in DNS that enable email authentication, and these played a significant role in our experimental study.

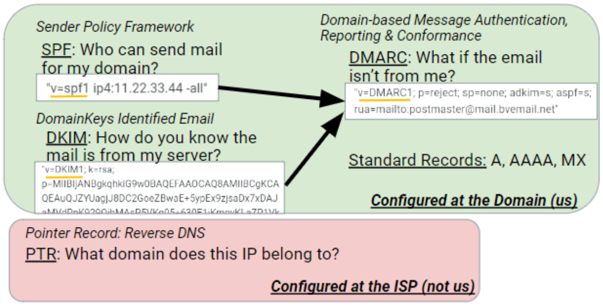

Sender Policy Framework
SPF: Who can send mail for my domain?
"v=spf1 ip4:11.22.33.44 -all"
DomainKeys Identified Email
DKIM: How do you know the mail is from my server?
"v=DKIM1; k=rsa;
p=MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCA
QEAuQJZYUagjJ8DC2GoeZBwaE+5ypEx9zjsaDx7xDAJ
Domain-based Message Authentication, Reporting & Conformance
DMARC: What if the email isn't from me?
"v=DMARC1; p=reject; sp=none; adkim=s; aspf=s; rua=mailto:postmaster@mail.bvemail.net"
Standard Records: A, AAAA, MX
Configured at the Domain (us)
Pointer Record: Reverse DNS
PTR: What domain does this IP belong to?
Configured at the ISP (not us)

Figure 2: DNS Records Relevant to Email Authentication

**Sender Policy Framework** (SPF) records are used by domain administrators to specify which hosts or third-party domains can send emails using their domain name [16]. Simple records include a set of hosts with a qualifier (pass, fail, soft fail, neutral).

**DomainKeys Identified Mail** (DKIM) records list public keys associated with the sending domain [21]. The sending servers use the corresponding private keys to cryptographically sign outgoing messages, allowing the recipients to verify a message's origin by comparing the signature to the domain's entry. RSA-SHA256 is commonly used as the signing algorithm.

**Domain-based Message Authentication, Reporting, and Conformance** (DMARC) records are used by a domain to specify handling policies for mail received using their domain name [22]. As illustrated in figure 2, DMARC records often create policies based on SPF and DKIM. DMARC is intended to allow the sender to establish criteria for message conformance and specify actions recipients should take if the message doesn't meet the stated conformance requirements. This usually includes handling instructions such as accept or reject and email addresses for feedback reports.

**Pointer Records** (PTR) are configured in a DNS lookup zone usually managed by the owner of the IP space. Whereas DNS Address (A) records associate a hostname with an IP address, PTR records are used for Reverse-DNS (rDNS), in which a lookup finds the associated hostname. Email providers sometimes use the presence and agreement of the PTR for the email-sending IP address to determine whether the sender is likely to be legitimate or not [13].

## 3 Finding a Connection

We first needed to find an ISP connection to test from. Because we are primarily interested in sending email and don't necessarily need to receive it, we tested viability from the local network using telnet on port 25 to portquiz, a website service that listens on all ports [24].

### 3.1 Selecting an ISP

#### 3.1.1 Xfinity

Our first candidate was the most readily available network to us, a team member's home connection through Xfinity. We discovered that TCP Port 25 was blocked. However, we initially thought we might be able to circumvent this by using an alternative port to transfer the email to remote servers. We discovered, for reasons described in section 2, that this solution was not likely to be viable with our constraints, so we pursued having the port unblocked. We were initially directed to the Xfinity Customer Security Assurance team, who then pointed us to Xfinity Internet Customer Support. After we were unable to make progress with customer support, we reached out to The Office of Tom "K" Karinshak, the Xfinity Chief Customer Experience Officer. After understanding our request and the intent, the representative informed us that Xfinity does not unblock port 25 for residential customers, citing FTC recommendations.

#### 3.1.2 Cloud Providers

This left us needing a testing platform, so we expanded the scope slightly to include cloud service providers, specifically in the context of individual user accounts. We initially explored Amazon Web Services (AWS), which blocks port 25 by default but has a published guide describing how a customer can request the port be unblocked for a particular virtual networking instance [28]. However, when we followed this procedure, our request was not approved. We did not receive meaningful feedback on why this happened, but we effectively removed AWS from the list of possibilities. Likewise, we explored other popular cloud computing providers. Microsoft Azure has a process for enabling the port, but it only applies to enterprise accounts; regular user accounts are not eligible to access port 25 [4]. Google Cloud documentation explicitly states they do not permit virtual machines to access the port [29]. While Linode advertises a process for enabling the port, we did not pursue this because of the associated cost [23].

#### 3.1.3 Regional WISP

We finally found a suitable testing connection via a small Wireless Internet Service Provider (WISP) in central Georgia. WISPs use radio equipment to deliver internet to customers in rural areas who usually do not have access to wired broadband like DSL or cable [1]. The particular WISP we used has a small customer footprint but permits traffic on port 25 by default. However, using this solution introduced some additional limitations we address in section 6.

### 3.2 ISPs and Port 25

Why do many providers block port 25 on their networks? We the most significant supporting documentation regarding their decision [30]. They refer to guidance from the Messaging Malware Mobile Anti-Abuse Working Group (MAAWG) and Internet Engineering Task Force (IETF), as well as from industry groups and the FTC. Some of their listed references are summarized here.

**MAAWG Recommendation: Managing Port 25 for Residential or Dynamic IP Space Benefits of Adoption and Risks of Inaction** [10], originally published in 2005, overviews some of the risks of unsecured residential networks and suggestions for best practices and their benefits. The most relevant of these include using port 587 for authenticated email submission per IETF RFC 2476 and blocking port 25 from hosts on the ISP network unless explicitly allowed.

**IETF RFC 4409 - Message Submission for Mail** [18] replaced RFC 2476 in 2006 with updated guidance for email submission and has since been replaced by RFC 6409. The most significant purpose of these RFCs are to standardize email submission and promote moving submission from MUAs to MSAs onto port 587, which requires authentication, to separate it from SMTP email relay on port 25.

**IETF RFC 5068 - Email Submission Operations: Access and Accountability Requirements** [12] presents a summary of best practices for enterprises and ISPs regarding email submission. It primarily supports the correct use of port 587 and implies that port 25 should not be used for submission but explicitly avoids recommending blocking port 25 across networks.

**Operation Spam Zombies: Letter to ISPs** [3] is a short best practice document developed in 2005 by a multinational partnership of government agencies led by the FTC to combat spam zombies. Their recommendations to ISPs include blocking port 25 except for authenticated client traffic.

In addition to shifting email submission from port 25 to port 587, many ISPs also have policies against users running servers on their residential internet connection, which use dynamic IPs. In combination, these two considerations make blocking port 25 an easily defensible decision. While Xfinity is one of the few providers to outline the reasoning behind its decision to prevent its customers from using port 25, other ISPs certainly do the same. Many common US service providers block the port per their website or support forums, including [2, 25, 30, 31]:

- Verizon
- AT&T
- NetZero
- Cox
- EarthLink
- Starlink
- CenturyLink
- HughesNet

# 4 Test Setup

## 4.1 Our Configuration

### 4.1.1 Server

We then proceeded to configure our test system. We decided it would be easiest to ensure well-configured emails were being sent and keep track of responses if we set up a server. Initially, we worked on installing Postfix and Dovecot on Ubuntu Server manually. However, we found an email stack called iRedMail that provides the essential functions in a mostly pre-configured package. This significantly reduced the time required to establish a working server and allowed us to easily incorporate additional quality-of-life tools, such as the iRedMail administration web interface.

This stack still uses Postfix as the MTA and Dovecot as the MSA. It also includes other common open-source tools, including RoundCube for webmail, Amavisd, ClamAV, SpamAssassin, and Fail2ban for security, and others that can be seen in more detail in their documentation [14].

### 4.1.2 Domain

We used Google Domains as the registrar for our testing domain. We also used their included DNS service and web interface to configure and change our domain's records. During our initial setup and configuration phase, we configured the DNS records to include SPF, DKIM, and DMARC, so we could confirm that our emails correctly matched our configuration during testing.

### 4.1.3 Verification

We verified our test setup by checking our configuration using several online tools. Google Admin Toolbox's Check MX provided a basic check to ensure our DNS records were all input correctly and our selectors were working as expected [9]. We confirmed these results using MXToolbox, which provides more detailed reporting [27]. Finally, we used mimecast's DMARC analyzer to certify that our email messages and DNS configuration matched [26]. During this verification stage, we found that different services handle records for subdomains differently. To mitigate this, we duplicated most of the authentication records for our top-level domain and our mail subdomain. With this addition, our DNS records and email configuration passed all tests.

### 4.1.4 Targets

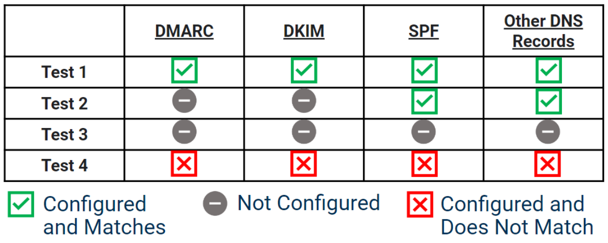
We selected free email providers to use as testing targets based mainly on the service's popularity. We created accounts for testing on eight providers:

- Google Gmail
- Yahoo Mail
- Yandex
- Proton
- Microsoft Outlook
- GMX
- Tutanota
- AOL Mail

We also intended to test Apple's iCloud email but eventually excluded it due to its requirement to register an Apple hardware device to access an account's email features.

## 4.2 Test Cases

With our testing platform established, we sent emails from our server to our targets. As seen in figure 3, we tested with four different DNS configurations to measure the email providers' policies. After sending emails for each one of the test sets, we updated the DNS records for the next test. We then waited for at least an hour, the Google DNS record timeout limit, before sending the next set of messages.



| | DMARC | DKIM | SPF | Other DNS Records |
|---|---|---|---|---|
| Test 1 | ☑ | ☑ | ☑ | ☑ |
| Test 2 | ⊖ | ⊖ | ☑ | ☑ |
| Test 3 | ⊖ | ⊖ | ⊖ | ⊖ |
| Test 4 | ☒ | ☒ | ☒ | ☒ |

☑ Configured and Matches ⊖ Not Configured ☒ Configured and Does Not Match

Figure 3: DNS Configurations for Tests

**Test 1: Fully Configured** In the first test, we applied SPF, DKIM, and DMARC in addition to standard DNS records such as MX and A. The message we sent fully complied with all of these records.

**Test 2: SPF Only** We removed DKIM and DMARC from our DNS records in the second test. However, the sent emails still matched the SPF record.

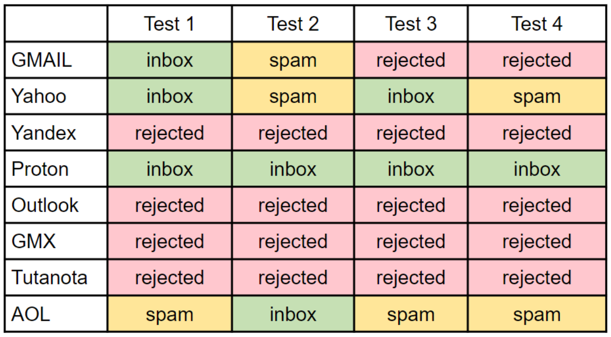
**Test 3: No DNS** For the third test, we removed all DNS records from our domain.

**Test 4: Incorrect** For the final test, we added all of the records for the domain again, including SPF, DKIM, and DMARC, similar to Test 1. However, this time the emails we sent did not match any of the records and violated the domain's policies.

## 5 Results

We classified our results for each test based on whether the message to each provider was accepted or not. If the email was accepted, we checked if the target user had it sorted into the inbox or spam folder. If the email was not accepted, or bounced, we examined the error message if one was provided.

### 5.1 Testing Round 1



| | Test 1 | Test 2 | Test 3 | Test 4 |
|---|---|---|---|---|
| GMAIL | inbox | spam | rejected | rejected |
| Yahoo | inbox | spam | inbox | spam |
| Yandex | rejected | rejected | rejected | rejected |
| Proton | inbox | inbox | inbox | inbox |
| Outlook | rejected | rejected | rejected | rejected |
| GMX | rejected | rejected | rejected | rejected |
| Tutanota | rejected | rejected | rejected | rejected |
| AOL | spam | inbox | spam | spam |

Figure 4: Round 1 Testing Results

After the first round of testing, four of the email providers accepted emails from our domain - three in the inbox and one in the spam folder. Three out of these four providers still accepted our Test 4 message that violated domain policies. The other half of the providers rejected the emails for all of our tests, regardless of configuration. In general, our results seemingly suggest domain email authentication, specifically DKIM, DMARC, and SPF records are not used as a primary measure of email trustworthiness. Most email providers either rejected all of the test emails or accepted all of them, with Gmail being the only exception. Providers seem to primarily rely on other heuristics or criteria to determine whether or not to accept incoming mail.

There were several interesting and unexpected results: Yahoo (Test 3) and AOL (Test 2) sorted some emails into the inbox after they filter a "better" email to spam. We are not sure why this occurred.

Proton placed all of the test messages into the user's inbox, including Test 4, although that message had a warning stating email did not match the domain's policies. This seemed strange to us, and we questioned if our results were representative of their policies.

We also received several error messages throughout our tests:

*Google* provided a different error for Test 3 and Test 4. The first (550 5.7.25) mentioned a missing PTR record, which was interesting because that was also absent during the first two tests. The second (550 5.7.26) stated the message failed to pass the domain SPF checks, and had been blocked to prevent spam.

*Yandex* issued the same error for every test (550 5.7.1), an undescribed policy rejection.

*Outlook* also issued the same error for every test (550 5.7.1) which stated that part of the ISP's network is on one of the Outlook blocklists (S3150). Presumably, this blocklist includes most or all dynamic or residential ISP space.

*GMX* issued identical error messages (554) for every test, stating the GMX server refused the connection. The information link included in the error listed no valid PTR record for the rejection.

*Tutanota* sent the same error message (450 4.7.1) for every test. The error states that they were unable to find the sender's reverse hostname, also indicating the missing PTR.

### 5.2 Testing Round 2

In order to confirm our test results, we later ran a second round of tests. The primary change in testing methodology for this iteration was a new email address was created for each test, where the same one was reused for all tests during the first round. The results were mostly the same for each provider, with a few additional observations or exceptions:

**AOL Mail** Test 1 was delivered to the inbox, Test 3 was filtered to spam. Tests 2 and 4 were not delivered, but also did not provide an error.

**Yahoo Mail** Test 2 was delivered to the inbox, Tests 1 and 4 were both delivered to spam. Test 3 was not delivered but also did not provide an error.

**Proton Mail** All tests were again delivered to the inbox, which suggests that the results from our first testing round were valid.

It's unclear if the undelivered mail items were silently dropped or were undelivered due to some technical failure. Regardless, the results overall seem to agree with the results from the first round of testing.

### 5.3 Investigating Proton

We initially thought the Proton results from the first round of testing were potentially anomalous or polluted somehow. For example, we considered if the email address we were using for testing somehow was added to some trusted senders list for the Proton target user. Alternatively, we considered if the filtering decisions were made for a sender, then cached for some period of time before they were refreshed. However, the second round of testing suggests that the first hypothesis is untrue and the warning label for the Test 4 email seems to indicate that the system did compare the message to the most recent DNS records.

We confirmed these results by using another domain we control with an SPF record that expresses that no senders for the domain are authorized. The email from the new domain was also accepted by Proton and placed into the inbox. When we configured this secondary domain's DNS records to match Test 4, it was handled the same way.

We reached out to Proton's support team, asking for insight into whether this behavior is intended and, if so, why they decided on a relatively permissive stance. Their response was, essentially, domain authentication for email is not a reliable measure because it is not uncommon for legitimate emails to fail authentication. This can happen for a variety of reasons, but the most frequent are improper forwarding or the use of third-party services for mass mailing.

## 6 Limitations and Future Work

Our results are interesting but non-authoritative at this stage. There were several limitations to our testing and many corresponding directions the experiment could be further expanded.

**More Vantage Points** Due to our difficulty finding viable ISP connections to test from, we were limited to testing from a single ISP and IP. Ideally, we would be able to conduct similar tests from multiple IP spaces owned by several residential providers in order to add confidence to our results. It also may be beneficial to repeat the tests with diverse domains, but financial considerations prevented us from doing so for this paper.

**Expand Tested Configurations** We only tested a limited subset of the total potential configurations for enumerating email providers' policy based filtering. A notable limitation, as a result of the WISP's administration policy, is we were unable to conduct any tests that included a valid PTR record. However, this is a common policy for residential ISPs, so not including PTR records is reasonable given our total project scope.

**Test Against Blocklists** Sending from domains associated with various blocklists would allow us to compare those results against the control of a clean address like the one we used for testing during this experiment.

**Include IPV6** We hoped to do testing from IPv4 and IPv6 addresses separately to compare the results; however, the WISP is IPv4 only, so we were unable to conduct any testing using IPv6. It would be particularly interesting to examine the two in the context of email blocklists.

## 7 Conclusion

In this paper we conducted an experimental study in which we demonstrated that it is possible to successfully send email from a residential internet connection into the inboxes of users of some popular free email providers. However, this success was pyrrhic, as the frequency of ISPs blocking port 25 means leveraging compromised home devices en mass to send spam directly is still difficult. We also observed that email domain authentication tools seem to not be a strong determinant for whether an email is delivered or not.

## Acknowledgments

Thanks to Prof. Paul for being supportive, even when we were not making headway. We also thank Proton support for being responsive and open to our inquiries.

## References

[1] AirFi. About airfi. https://www.airfi.today/about.

[2] CenturyLink. Understanding port 25 filtering. https://www.centurylink.com/home/help/internet/email/understanding-port-25-filtering.html.

[3] Federal Trade Commission. Ftc, partners launch campaign against spam zombies. https://www.ftc.gov/news-events/news/press-releases/2005/05/ftc-partners-launch-campaign-against-spam-zombies.

[4] Microsoft Community. Troubleshoot outbound smtp connectivity problems in azure. https://learn.microsoft.com/en-us/azure/virtual-network/troubleshoot-outbound-smtp-connectivity.

[5] Multiple Contributers. Shadow banning. https://en.wikipedia.org/wiki/Shadow_banning#cite_note-TechCrunch_2013_Hacker_News-34.

[6] Sarah Delany. Using case-based reasoning for spam filtering. *Technological University Dublin*, 12 2022.

[7] Carlos Fenollosa. After self-hosting my email for twenty-three years i have thrown in the towel. the oligopoly has won. https://cfenollosa.com/blog/after-self-hosting-my-email-for-twenty-

three-years-i-have-thrown-in-the-towel-the-oligopoly-has-won.html.

[8] Scott Figario. Do you know the history of spam? https://www.socketlabs.com/blog/know-history-spam/#:~:text=The%20history%20of%20spam%20can,offers%20sent%20to%20wealthy%20Americans.

[9] Google. Google admin toolbox: Check mx. https://toolbox.googleapps.com/apps/checkmx/.

[10] Messaging Anti-Abuse Working Group. Managing port 25 for residential or dynamic ip space benefits of adoption and risks of inaction. https://www.m3aawg.org/sites/default/files/document/MAAWG_Port25rec0511.pdf.

[11] Sally Hambridge and Albert Lunde. DON'T SPEW A Set of Guidelines for Mass Unsolicited Mailings and Postings (spam*). RFC 2635, June 1999.

[12] Carl Hutzler, Dave Crocker, Eric P. Allman, Pete Resnick, and Tony Finch. Email Submission Operations: Access and Accountability Requirements. RFC 6409, November 2007.

[13] Mail & Media Inc. Gmx postmaster: Email policy. https://postmaster.gmx.com/en/email-policy.

[14] iRedMail. iredmail docs. https://docs.iredmail.org/index.html.

[15] Chris Kanich, Christian Kreibich, Kirill Levchenko, Brandon Enright, Geoffrey M. Voelker, Vern Paxson, and Stefan Savage. Spamalytics: An empirical analysis of spam marketing conversion. *Commun. ACM*, 52(9):99–107, Sep 2009.

[16] Scott Kitterman. Sender Policy Framework (SPF) for Authorizing Use of Domains in Email, Version 1. RFC 7208, April 2014.

[17] Dr. John C. Klensin. Simple Mail Transfer Protocol. RFC 5321, October 2008.

[18] Dr. John C. Klensin and Randall Gellens. Message Submission for Mail. RFC 4409, April 2006.

[19] Dr. John C. Klensin and Randall Gellens. Message Submission for Mail. RFC 6409, November 2011.

[20] Lyudmila Kovalenko. How to send millions of emails and avoid spam filters. https://altcraft.com/blog/how-to-send-millions-of-emails-and-avoid-spam-filters.

[21] Murray Kucherawy, Dave Crocker, and Tony Hansen. DomainKeys Identified Mail (DKIM) Signatures. RFC 6376, September 2011.

[22] Murray Kucherawy and Elizabeth Zwicky. Domain-based Message Authentication, Reporting, and Conformance (DMARC). RFC 7489, March 2015.

[23] Linode. Running a mail server. https://www.linode.com/docs/guides/running-a-mail-server.

[24] Marc Maurice. Outgoing port tester. http://portquiz.net/.

[25] mexx4. Smtp port 25 blocked on starlink. https://www.reddit.com/r/Starlink/comments/pclc98/smtp_port_25_blocked_on_starlink/.

[26] mimecast. dmarc analyzer. https://www.dmarcanalyzer.com/.

[27] Inc MxToolBox. Mxtoolbox: Supertool. https://mxtoolbox.com/SuperTool.aspx.

[28] AWS Support. How do i remove the restriction on port 25 from my amazon ec2 instance or aws lambda function? https://aws.amazon.com/premiumsupport/knowledge-center/ec2-port-25-throttle/.

[29] Google Cloud Support. Sending email from an instance. https://cloud.google.com/compute/docs/tutorials/sending-mail.

[30] Xfinity Support. Why is port 25 for email submission not supported? https://www.xfinity.com/support/articles/email-port-25-no-longer-supported.

[31] Warren. Having trouble sending or receiving email? try switching to port 587. https://community.hughesnet.com/t5/About-the-Community/Having-trouble-sending-or-receiving-email-Try-Switching-to-Port/m-p/7345.