# Unfiltered: Measuring Cloud-based Email Filtering Bypasses

Sumanth Rao
UC San Diego
La Jolla, CA, USA
svrao@ucsd.edu

Enze Liu
UC San Diego
La Jolla, CA, USA
e7liu@ucsd.edu

Grant Ho
University of Chicago
Chicago, IL, USA
grantho@uchicago.edu

Geoffrey M. Voelker
UC San Diego
La Jolla, CA, USA
voelker@cs.ucsd.edu

Stefan Savage
UC San Diego
La Jolla, CA, USA
savage@cs.ucsd.edu

## ABSTRACT

Email service has increasingly been outsourced to cloud-based providers and so too has the task of filtering such messages for potential threats. Thus, customers will commonly direct that their incoming email is first sent to a third-party email filtering service (e.g., Proofpoint or Barracuda) and only the "clean" messages are then sent on to their email hosting provider (e.g., Gmail or Microsoft Exchange Online). However, this loosely coupled approach can, in theory, be *bypassed* if the email hosting provider is not configured to *only* accept messages that arrive from the email filtering service. In this paper we demonstrate that such bypasses are commonly possible. We document a multi-step methodology to infer if an organization has correctly configured its email hosting provider to guard against such scenarios. Then, using an empirical measurement of *edu* and *com* domains as a case study, we show that 80% of such organizations making use of popular cloud-based email filtering services can be bypassed in this manner. We also discuss reasons that lead to such misconfigurations and outline challenges in hardening the binding between email filtering and hosting providers.

## CCS CONCEPTS

• **Information systems** → **Email**; • **Security and privacy** → **Vulnerability management**.

## KEYWORDS

Email, Security, Measurement, Filtering, Bypass, SMTP.

**ACM Reference Format:**
Sumanth Rao, Enze Liu, Grant Ho, Geoffrey M. Voelker, and Stefan Savage. 2024. Unfiltered: Measuring Cloud-based Email Filtering Bypasses. In *Proceedings of the ACM Web Conference 2024 (WWW '24), May 13–17, 2024, Singapore, Singapore.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3589334.3645499

## 1 INTRODUCTION

Over the last decade, a range of economic incentives have driven enterprises to abandon key self-hosted services and outsource these functions to third-party cloud-based service providers. This trend has encompassed services including storage (e.g., Dropbox, GDrive, etc.), backup (e.g., Backblaze), domain names (e.g., Amazon Route 53, Cloudflare DNS), productivity applications (e.g., GSuite, Microsoft 365), web hosting (e.g., Cloudflare, AWS) and, importantly for this paper, email (e.g., Gmail and Microsoft Exchange Online). However, there is no established or standardized implementation or protocol for composing such cloud services, and thus each situation is managed in an application-specific ad hoc manner. In this paper we focus on a simple example of this situation: the interaction between cloud-based email services and email filtering services.

While the third-party email hosting providers commonly used by enterprises provide native filtering capabilities, many organizations prefer to supplement these capabilities with specialized third-party filtering services, such as those offered by Proofpoint [53] or Barracuda [5].[1] As with their on-premises appliance predecessors, such services offer enhanced policy control, more advanced security features (e.g., URL-rewriting and attachment "detonation"), rich reporting capabilities, and market themselves as being singularly focused on defending against the latest email-borne threats.

Implementing this filtering step in the cloud requires a mechanism to manage the flow of inbound email — funneling mail *first* to the filtering service and *then* to the email provider. While there are a range of ways such a capability could be implemented in principle, in practice the common mechanism is to configure a domain's DNS "Mail Exchanger" (MX) record to direct incoming traffic to the email filtering service and then configure that service to deliver the filtered email stream to the domain's email hosting provider. Although this procedure ensures that filtered email is ultimately delivered, it does not guarantee that delivered email has been filtered. Indeed, a clever adversary could identify the server used by a domain's email hosting provider and send malicious mail *directly* to them, thus bypassing the third-party filtering (and the security benefits it provides).[2] There are a number of ad hoc measures that an enterprise might take to protect against such actions (e.g., rejecting email from IP addresses not operated by the email filtering service), but such defensive configurations are neither required for correct operation nor are they externally visible to any outside auditor.

This paper investigates the deployment of cloud-based email filtering services via two primary contributions:

This work is licensed under a Creative Commons Attribution International 4.0 License.

*WWW '24, May 13–17, 2024, Singapore, Singapore*
© 2024 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-0171-9/24/05.
https://doi.org/10.1145/3589334.3645499

[1] Liu et al's 2021 study of email providers shows that 13% of the Alexa Top 1k domains made use of one of these two filtering services for this purpose [34].
[2] Several email filtering services are aware of this issue and have noted such possibility in their documentation.

- Through careful controlled trials, we have developed and validated a multi-step measurement procedure to infer a domain owner's choice of email hosting provider, email filtering service, and the integrity of the binding between the two (i.e., whether the filtering relationship is "bypassable").
- Using this technique, we have conducted case studies focused on auditing 673 `edu` domains and a sample of 928 popular `com` domains that use the 15 most prevalent cloud-based email filtering services (e.g., Proofpoint, Barracuda, Cisco, Mimecast). We show that 80% of these domains do not protect the integrity of the email delivery path and therefore their filtering can be trivially bypassed.

Finally, based on our analysis, we describe the challenges and tradeoffs involved in addressing this problem, which represents a special case of an overall challenge in architecting composition between third-party cloud services.

## 2 BACKGROUND

We begin by reviewing the SMTP protocol involved in email transmission. We then provide an overview for email delivery in the presence of a cloud-based email filtering service, and describe how a clever attacker might bypass such a setup without appropriate countermeasures. Finally, we discuss how email hosting providers can be properly configured to prevent such bypassing threats.

### 2.1 Simple Mail Transfer Protocol

The simple mail transfer protocol (SMTP) is a family of protocols that governs the transmission of email messages [28], including email forwarding and delivery. All protocols in the SMTP family are text-based and follow a similar session-based model. Figure 1 depicts a typical SMTP session between a client (C) and a server (S). The session starts when the client initiates a successful connection with the server, and they both announce their identities in BANNER and EHLO messages. Next, the client specifies the email address of the sender (e.g., `decision@webconf.org`) with the MAIL FROM command. The server responds with a 250 message code on success or a specific error code on failure. The client then specifies the email address and information about the recipient (e.g., `author@univ.edu`) in the RCPT command (hence referred to as the RCPT address), and the server again acknowledges. Next, the client sends a DATA command and the contents of the email message, and ends with a period ("."). The server acknowledges and delivers the message to the recipient's mailbox. While Figure 1 depicts a traditional case in which the server is hosted by `univ.edu`, it is increasingly common that this email service is instead outsourced to Gmail or Microsoft Exchange Online [34].

| | |
|---|---|
| **S**: 220 univ.edu SMTP Service Ready | **BANNER** |
| **C**: EHLO webconf.org | |
| **S**: 250 univ.edu greets webconf.org | **EHLO** |
| **C**: MAIL FROM: <decision@webconf.org> | |
| **S**: 250 OK | **MAIL** |
| **C**: RCPT TO:<author@univ.edu> | |
| **S**: 250 OK | **RCPT** |
| **C**: DATA | |
| **C**: From: decision@webconf.org | |
| **C**: To: author@univ.edu | |
| **C**: Subject: Paper Decision | **DATA** |
| Decision is enclosed in this email. | |
| . | |
| **S**: 250 OK. Queued mail for delivery. | |
| **C**: QUIT | **QUIT** |

**Figure 1: A typical SMTP session between a client (C) and a server (S) that handles mail for `univ.edu`.**

### 2.2 Email Delivery with Filtering Services

Organizations have also increasingly adopted cloud-based email filtering services to defend against various email-based threats [34]. As illustrated in Figure 2, these cloud services act as gateways between the Internet and organizational email servers, expanding the process of email delivery beyond one SMTP session. First, the sender uses their Mail User Agent (MUA) to craft and submit a message to their email server (step 1).[3] The sender's server then identifies the recipient's server by querying the DNS MX record associated with the recipient's domain (step 2). If the recipient's organization uses a cloud-based email filtering service, the recipient's MX record points to an email server hosted by the filtering service. The sender's server then initiates an SMTP session with the filtering service's server and sends the email to them (step 3). After processing (e.g., spam filtering and URL rewriting), the filtering service then forwards the email to the recipient's server (step 4). The recipient can then retrieve the email from their organization's mail server and display the message using their MUA (step 5).

### 2.3 Bypassing Email Filtering Services

Filtering works as intended when the sender follows the normal email transmission flow (i.e., querying the recipient's MX record and sending to its designated mail server). However, as mentioned earlier, a clever adversary might bypass the filtering service by directly sending email to the recipient's server ("direct delivery") after inferring the recipient's email hosting provider. As depicted in Figure 2, instead of performing steps 2–4, the adversary directly delivers email to the recipient's mail server, bypassing the filtering service and any protection provided by it. Analyzing the extent to which organizations using cloud-based filtering services are susceptible to this kind of bypass attack is the focus of our work.

### 2.4 Preventing Bypass

To prevent such bypasses from happening, organizations can harden the binding between their filtering service and their email server by configuring their server to only accept messages from the filtering service. Surveying the documentation of major filtering services and hosting providers, we identified three email providers that have a mechanism to appropriately restrict inbound email delivery:

[3]With web-hosted third-party mail services, such as provided by Gmail, the MUA and email server may in fact be part of the same service offering.

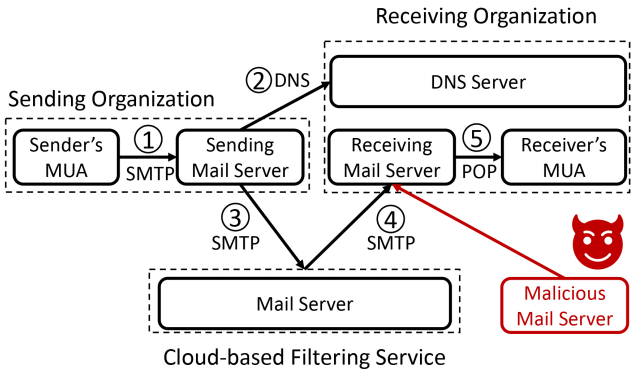


**Figure 2: The email transmission flow when a recipient's organization uses a cloud-based filtering service, and how an attacker can bypass the filtering service for insecurely configured organizations.**

Gmail, the email service for Google Workspace; Microsoft Exchange Online (hence referred to as "Exchange Online"), the email service for Microsoft 365; and Zoho Mail (hence referred to as "Zoho"). All three providers share the same underlying idea: using an "allow list" to only accept inbound email originating from IP addresses associated with the appropriate filtering service.

While the underlying idea is the same across all three providers, the configuration syntax varies. For Gmail and Zoho, organizations first specify the IP addresses of their filtering service in a dedicated "inbound gateway IP list", and then enable a separate feature that rejects all email not from gateway IPs. By contrast, Exchange Online does not have a dedicated configuration option for this purpose. Instead, organizations can specify the IP addresses of a filtering service and associated rules using a "connector" [39], a general mechanism for customizing email routing [6, 46, 58].[4]

## 3 METHODOLOGY

While the *potential* for such email filter bypassing is evident in the design and documentation of these services, whether such vulnerabilities exist in practice is a separate empirical question. To explore this further, we must first identify those organizations using third-party mail filtering services, determine the underlying mail hosting provider to which their filtered mail will be delivered, and then establish, via measurement or inference, the integrity of their delivery path (i.e., whether such a bypass is feasible).[5]

### 3.1 Identifying Mail Filtering Service

Third-party mail filtering services are designed to be the first point of contact in a domain's mail delivery path. While this goal is achieved by setting a domain's MX record to direct all inbound messages to the filtering service, it can be implemented in a number of ways in practice. For example, `ucsd.edu` might set its MX record to `xxx.gslb.pphosted.com` (a domain operated by Proofpoint), or it might point to `inbound.ucsd.edu` which further resolves via A record to an IP address in the prefix `148.163.128.0/19` (operated by Proofpoint), or it might use even more complex combinations of CNAME settings and multiple levels of name resolution or proxying.

In prior work on identifying email services, Liu et al. [34] show how the combination of MX record, A record, TLS certificate (for domains accepting TLS for SMTP mail delivery), SMTP banner and protocol response can obtain a high-confidence assessment of the organization accepting mail delivery for a domain. Using a variant of this approach we develop "signatures" for 15 leading email filtering services: Proofpoint, Mimecast, Cisco (aka Ironport), Barracuda, TrendMicro, Broadcom (aka Symantec), Trellix (formerly FireEye), Sophos, Cloudflare, Fortinet, N-able (formerly SolarWinds MSP), Forcepoint, AppRiver, Spamhero and HornetSecurity.[6]

We then apply this approach to a corpus of registered `edu` and `com` domains. Since EDUCAUSE (the registry for `edu`) does not publish its DNS zone files, we construct this list using `edu`-containing X.509 certificates collected by Censys [13]. For our `com` set, we use the 50k most popular domains as identified by Google's Chrome User Experience Report (CrUX) [19]. We remove domains that do not have valid MX records (i.e., do not accept mail) and further extract the subset that make use of email filtering services (using the signatures we develop). These steps produce a corpus of 889 `edu` domains and 1,429 `com` domains that make use of one of these 15 services (15–17% of each corpus). Consistent with prior findings, Proofpoint is the dominant filtering service in our data, followed by Barracuda, Mimecast and Cisco which together serve 89% of the domains using third-party email filtering.

### 3.2 Inferring Mail Hosting Provider

A domain's use of an email filtering service can be measured directly, but *where* such a service subsequently delivers the filtered mail is not directly visible. Since few domains publicize which mail hosting provider they use, we have developed measurement workflows to infer with high confidence if a filtered domain uses Google, Microsoft or Zoho as their backend email hosting provider.[7] By analyzing the documentation of these three providers, along with insights gleaned from mail administrator forums, and by empirically creating and testing our own subscriptions to these services, we establish that all three providers expose some externally visible state when an organization has a valid subscription (and, crucially, this state is not evident when the subscription is deleted or defunct).

In particular, it is well-documented that when an organization has a valid Gmail subscription, Google automatically creates a `postmaster` and an `abuse` email address associated with the organization's domain name [18, 20, 67]. Zoho similarly creates default `postmaster` and `abuse` addresses for each domain with a valid subscription [36]. While Exchange Online does not automatically create default email addresses, it automatically creates a uniquely-formatted A record under the `mail.protection.outlook.com`

[4]The generality of this mechanism can be confusing and, perhaps as a result, we observe that some filtering service documentation incorrectly instructs their customers to implement insecure configurations.

[5]Our code that identifies underlying mail hosting provider and the mail path integrity is available upon request.

[6]This list captures the leading mail security providers in two industry reports on the sector [16, 21] as well as a few others that appeared repeatedly in our data.

[7]We select these three because they are the major mail hosting providers that provide a mechanism to secure the mail delivery path; absent such a mechanism, *all* other mail hosting providers are *de facto* "bypassable". Moreover, as identified in previous work, Gmail and Exchange Online dominate the email hosting market — implementing the mail backend of roughly 40–45% of well-trafficked `com` domains [34].

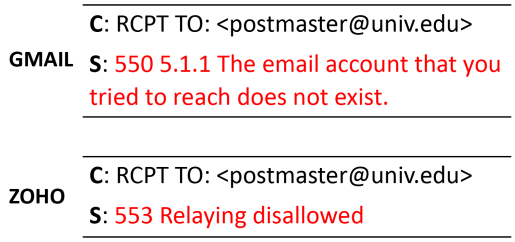


**Figure 3: Gmail returns a 550 error code, and Zoho returns a 553 error code when the recipient address does not exist.**

subdomain [2, 40, 42]. For example, if `univ.edu` has contracted with Exchange Online, then Microsoft will create an A record for `univ-edu.mail.protection.outlook.com` [42, 48].[8]

Critically, each of these pieces of state is externally testable. Thus, we can infer whether a domain `foo.com` is associated with a valid Gmail subscription by connecting to Gmail's SMTP servers and specifying the `postmaster@foo.com` address as a parameter to the RCPT command. As shown in Figure 3, if `foo.com` is not hosted by Gmail then it will return a 550 error code, otherwise it will return an OK or more specific error (discussed later). Similarly, Zoho's SMTP server will return a 553 error code if the domain does not have a current subscription, and a 250 OK code otherwise. Note that none of these tests require completing an SMTP transaction and thus do not send messages to the associated accounts. For Exchange Online, we can infer that a domain is associated with a valid subscription if the appropriate DNS record exists (e.g., `foo-com.mail.protection.outlook.com` for `foo.com`).

While this inference technique is both easy to perform and accurate, it is unable to distinguish between an organization that makes active use of a mail service from one which merely has an active subscription (e.g., an organization that has a Google Workplace subscription, but uses another provider for email). To avoid implicit bias from this effect, we adopt a conservative approach and only consider those domains showing evidence of active email use. We rely on the Sender Policy Framework (SPF), a widely-deployed email protocol designed to help prevent attackers from spoofing email. With SPF, a domain publishes a DNS TXT record specifying the list of domains and IP addresses authorized to send email on its behalf [27]. When actively using a third-party email hosting provider, organizations typically list the IP addresses of their provider in the SPF record (otherwise mail sent via the provider will be rejected or sent to spam folders by many recipients). To collect and incorporate this information, we use ZDNS [26] to parse and recursively query each domain's SPF record as needed (Appendix A).

After this filtering, 673 `edu` domains and 928 `com` domains remain: those actively making use of one of the three mail hosting providers *and* using one of the 15 mail filtering services (see Table 1).[9] It is this set of domains that we test for "bypassability".

[8]Also, as per Microsoft's documentation, organizations can optionally create a CNAME record that enables certain email clients (like Outlook) to automatically discover the Exchange server used and configure themselves correctly [38]. This CNAME record can similarly be used to infer the use of Exchange Online, as prior studies have done [35].
[9]The distribution of mail filtering services in this conservatively filtered set is consistent with the same distribution in the original corpus, suggesting that there is no correlation between the active use of the service and the choice of mail filtering service provider.

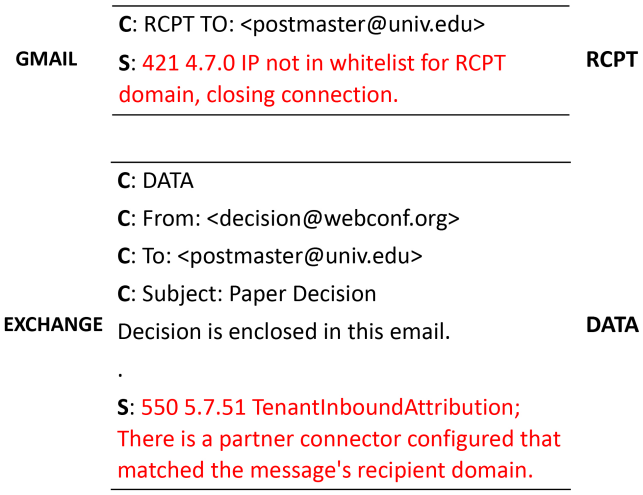


**Figure 4: Gmail returns a 421 error with a correctly configured protective "gateway", and Exchange Online returns a 550 error with a correctly setup protective "connector".**

## 3.3 Inferring Mail Path Integrity

Our goal is to understand whether a third-party email filtering service can be bypassed by sending directly to a domain's backend email provider. This question is determined entirely by the customer's configuration of the backend email provider: whether it will accept inbound email from any party, or if it will only accept such messages from the domain's filtering service.

This configuration behavior manifests itself when an unauthorized party (i.e., from an IP address not belonging to the mail filtering service) initiates an SMTP transaction with the mail provider and attempts to send mail to *valid addresses* in the domain. Based on systematic empirical testing, we have determined that securely-configured domains hosted by Gmail and Zoho will reject such an email during the RCPT stage of a session, while Exchange Online will reject during the DATA stage. Figure 4 illustrates this behavior for Gmail and Exchange Online. Messages are addressed to valid accounts in the domain, but the mail hosting provider is configured to only accept inbound mail from a specific filtering service. In this situation, Gmail returns a 421 error code while Exchange Online returns a 550 error code. Conversely, if the domain owner has not configured such inbound mail restrictions, then standard "250 OK" responses will be returned. A similar test distinguishes securely-configured Zoho-hosted domains. By connecting with each domain's backend mail servers and conducting such integrity tests we can infer whether their mail delivery path is secure or if it can be bypassed.[10]

For Gmail and Zoho, this integrity test is trivial to perform since, by default, there are well-known valid addresses (i.e., `postmaster` and `abuse`) for each hosted domain. However, Exchange Online has no such defaults and thus for this service we are forced to rely on heuristics. Further complicating testing, Exchange Online's integrity check takes place during the DATA command. Hence, if a bypass succeeds, a message will be delivered to the recipient. Since

[10]Note that Gmail and Zoho SMTP servers use well-known DNS addresses that accept mail traffic for all customers, while domains hosted by Exchange Online receive inbound mail via a unique domain-specific address as described earlier.

we wish to avoid imposing any undue burden on regular users of such email services, we cannot simply probe using common names or addresses obtained via search engines.

Instead, we first note that Exchange Online has a per-domain option, Directory-Based Edge Blocking [45], which causes the server to reject invalid addresses up front. However, if this feature is not enabled, then the integrity test can be performed using invalid addresses. Thus, we probe Exchange Online-hosted domains using a 25-character randomly generated alphanumeric email address in the RCPT command. If this address is accepted, then the invalid address blocking feature is disabled and this address will serve as a *valid address* for the purpose of integrity testing. If not, we then resort to blindly probing using a set of well-known *administrative addresses* (e.g., postmaster, admin, info, etc) as identified by Bennett et al. [8]. If any of these addresses are accepted in the RCPT command we then proceed with the previously described integrity test. Otherwise, we record the integrity of the domain as unknown. Only in one particular case, where an Exchange Online-hosted domain blocking invalid addresses has configured one of the well-known administrative email addresses *and* has not correctly configured the mail service to limit the bypass of the mail filtering service, will we end up delivering an email (discussed further in Section 5).

### 3.4 Limitations

Our methodology is based on assumptions that are well-suited to standard modes of use, but may fail in certain edge cases.

First, the presence or absence of an email provider's domain or IP addresses in an organization's SPF record is not a *guaranteed* indicator of its use (or not) of the provider. For example, an organization might have migrated to another provider and not yet updated its DNS record. Similarly, an organization might use Gmail yet decide to route all outbound email through its filtering service (and thus only include the filtering service in its SPF record). Our analysis would exclude such domains, even if vulnerable, because we cannot determine their email provider. While we believe such situations are atypical today, that might change in the future.

There are similarly rare edge cases around the configuration of email providers' inbound mail filtering. While our methodology focuses on inferring the use of recommended best practice (according to mail provider and filtering service documentation) we have seen ad hoc configurations that attempt to achieve the same effect (e.g., using Exchange Online's transport rules [41] to silently filter inbound email). In such cases, we might mistakenly determine that a domain's mail filtering service is bypassable, even though such an ad hoc filtering solution in fact protects it.

Finally, it is possible to use mail filtering services with self-hosted email servers and these, as well, may not secure the mail delivery path from bypass. Such scenarios are outside the scope of our current approach and will not be captured by our methodology.

## 4 RESULTS

In this section, we characterize the 1,601 domains in our dataset: we show the distributions of the filtering services and email providers used, and examine the extent to which these domains allow an attacker to bypass their filtering service. We also describe our approach to validate these findings on a subset of the domains.

| Filtering Service | Domains | edu | com |
|---|---|---|---|
| Proofpoint | 720 (45%) | 213 (32%) | 507 (55%) |
| Barracuda | 283 (18%) | 245 (36%) | 38 (4%) |
| Mimecast | 254 (16%) | 69 (10%) | 185 (20%) |
| Cisco | 160 (10%) | 96 (14%) | 64 (7%) |
| TrendMicro | 43 (3%) | 8 (1%) | 35 (4%) |
| Sophos | 29 (2%) | 17 (3%) | 12 (1%) |
| Trellix | 18 (1%) | – | 18 (2%) |
| Cloudflare | 18 (1%) | 1 (0.1%) | 17 (2%) |
| AppRiver | 18 (1%) | 8 (1%) | 10 (1%) |
| Broadcom | 15 (0.9%) | 2 (0.3%) | 13 (1%) |
| ForcePoint | 14 (0.8%) | 3 (0.4%) | 11 (1%) |
| Fortinet | 14 (0.8%) | 5 (0.7%) | 9 (0.9%) |
| Hornetsecurity | 8 (0.5%) | 2 (0.3%) | 6 (0.6%) |
| N-able | 4 (0.2%) | 2 (0.3%) | 2 (0.2%) |
| Spamhero | 3 (0.2%) | 2 (0.3%) | 1 (0.1%) |
| Total | 1,601 (100%) | 673 (100%) | 928 (100%) |

**Table 1: The cloud-based email filtering services considered in this study and their prevalence in our data.**

| TLD | Exchange | Gmail | Zoho |
|---|---|---|---|
| edu | 607 (85%) | 107 (15%) | 0 (0%) |
| com | 745 (75%) | 241 (24%) | 4 (0.4%) |
| Total | 1,352 (79%) | 348 (20%) | 4 (0.2%) |

**Table 2: The number of domains inferred to use each of the three email providers in each of the TLDs. Since 6% of domains use two email providers, we include them in both counts of the providers they use (hence the total counts are slightly larger than in Table 1).**

### 4.1 Filtering Services & Email Providers

Table 1 shows the distribution of third-party filtering services used by the domains in our study. For each filtering service, the table shows the number of domains using the service and the percentage of all domains in each column that use the service. Noticeably, the market is dominated by a few companies: the top five services account for 90–93% of domains, with a long tail populated by the remaining ten companies. However, there is also considerable market variation between the two TLDs. Barracuda, for instance, is the most popular service among edu domains at 36%, but has considerably less market share in com and is ranked fourth with 4%.

Table 2 shows the number of domains that actively use the email providers. Since 6% of the domains (103/1,601) use two providers, we include them in both counts of the providers they use (hence the total counts are slightly larger than in Table 1). In both TLDs, Exchange Online is by far the most popular provider, with a slightly higher popularity among edu domains. Gmail is the other popular alternative, with Zoho having just four customers in our set.[11]

[11]Note that this characterization differs significantly from that in Liu et al. [34], likely because our test set is *conditioned* on the use of third-party mail filtering services.

## 4.2 Vulnerable Configurations

Table 3 characterizes the mail path integrity for the domains in our data. Each cell corresponds to a filtering service and email provider combination, and shows the number of misconfigured domains that allow direct bypass out of the total number of domains for that combination. For clarity we combine the results from both TLDs and exclude the Zoho results: the misconfiguration rates are much more correlated with the combination of filtering service and email provider than which TLD the domain is in, and the four domains that use Zoho are all misconfigured and vulnerable to bypass. As with Table 2, 6% of domains use two email providers and we infer the configuration status for each provider they use and count those configurations separately in these results. Recall from Section 3.3 that evaluating bypass for domains using Exchange Online requires sending email to a valid address at that domain. For 123 domains, we were unable to determine such a valid email address and thus we exclude those domains from these results.

Overall, the surprising result is that *the vast majority* of domains misconfigure their email provider when using third-party filtering services: 80% of domains in our data are misconfigured to allow email delivery that bypasses the filtering service. We also observe that domains misconfigure Gmail more often than Exchange Online: 88% of Gmail configurations allow bypass, while 78% of Exchange Online configurations do.

From our experience configuring email providers to use filtering services, we observe three potential reasons for this high misconfiguration rate: (1) missing documentation and poor awareness that careful configuration is necessary for security, (2) even when present, the documentation can be unclear about how to setup a secure configuration, and (3) concerns about deliverability, which lead to permissive (insecure) configurations.

For instance, Mimecast's documentation [47] for using the product with Google Workspace neither explicitly instructs the administrator to restrict inbound mail to gateway IP addresses, nor highlights the risk of not doing so. Indeed, 95% of domains using Mimecast with Gmail are vulnerable to bypass. Similarly, Cisco's documentation [10] omits any mention of locking down inbound mail IPs when used with Exchange. However, lack of documentation alone cannot fully explain these results. We find 87% of domains using Proofpoint with Gmail *also* have vulnerable configurations, in spite of Proofpoint's documentation [51] highlighting the potential risks of bypass and identifying the correct Gmail configuration option for preventing it.

Another potential contributor to this problem is confusion in vendor documentation. For example, Microsoft's connector documentation describes two options for identifying email sent from third-party organizations: "By verifying if the sender domain matches..." and "By verifying if the IP address of the sending server matches..." [44]. However, these options are easy to misunderstand; the second option *does not*, in fact, restrict inbound mail to come from a given IP address, but simply restricts the circumstances when the connector will be run.[12] As a result, such a configuration can be bypassed by directly delivering mail to the organization's Exchange server [61]. We suspect this is why contemporaneous documentation from TrendMicro [63] and Proofpoint [52] provide instructions that explicitly produce vulnerable configurations in this manner. However, this too can only provide a partial explanation. Mimecast [46] and Barracuda's [6] connector documentation correctly describe restricting inbound mail to gateway IP addresses as a "necessary" configuration step, yet misconfiguration rates, while better, are still quite high.

Finally, we have encountered anecdotal evidence that some domain operators explicitly choose to not restrict inbound email IPs due to concerns about how it may impair mail function. For example, reviewing online blogs, forums, and email filter documentation, we identify three such concerns surfaced about Gmail in this configuration: that its "Automatically detect external IP" feature interferes with whitelisting [29]; that Gmail may, at times, prevent delivery from the domain's own IP addresses [51]; and that features like Smart Banners and URL rewriting reportedly break DMARC/SPF, resulting in valid mail being labeled as spam [22]. Given these community experiences, an administrator configuring Gmail may conclude that it would be prudent not to restrict the IP ranges for incoming SMTP connections.

The combination of these three issues may collectively explain the high rate of vulnerable configurations, although it remains unclear which is most important in practice. We discuss these results and potential causes further in Section 7.

| Filtering Serv. | Exchange | Gmail | Total |
|---|---|---|---|
| Proofpoint | 415/541 (77%) | 152/175 (87%) | 567/716 (79%) |
| Barracuda | 186/244 (76%) | 26/27 (96%) | 212/271 (79%) |
| Mimecast | 113/171 (66%) | 69/73 (95%) | 182/244 (75%) |
| Cisco | 124/139 (89%) | 15/18 (83%) | 139/157 (89%) |
| TrendMicro | 30/30 (100%) | 10/12 (83%) | 40/42 (95%) |
| Sophos | 16/18 (89%) | 7/9 (78%) | 23/27 (85%) |
| Cloudflare | 8/8 (100%) | 10/14 (71%) | 18/22 (82%) |
| Trellix | 9/13 (69%) | 5/7 (71%) | 14/20 (70%) |
| AppRiver | 13/13 (100%) | 6/6 (100%) | 19/19 (100%) |
| ForcePoint | 11/13 (85%) | 1/1 (100%) | 12/14 (86%) |
| Fortinet | 13/14 (93%) | 1/1 (100%) | 14/15 (93%) |
| Broadcom | 10/12 (83%) | 3/3 (100%) | 13/15 (87%) |
| HornetSecurity | 2/8 (25%) | 1/1 (100%) | 3/9 (33%) |
| N-able | 3/3 (100%) | – | 3/3 (100%) |
| Spamhero | 2/2 (100%) | 1/1 (100%) | 3/3 (100%) |
| Total | 955/1,229 (78%) | 307/348 (88%) | 1,262/1,577 (80%) |

**Table 3: The integrity of the mail paths for the domains in our data set. For each combination of filtering service and email provider it shows the number and percentage of misconfigured domains. Domains with two email providers are counted twice, once for each provider. For 123 domains that use Exchange Online, we could not evaluate their configuration status and exclude them from the counts in this table.**

## 4.3 Validation

For a subset of the domains in our study, we used three techniques to validate the results of our inference methods. Table 4 shows the number of domains that we validated using each technique according to the filtering services used. In all cases, our validation

[12]The correct approach is to create a condition that always matches and *then* add a rule to block messages that don't arrive from gateway IP addresses.

| Filtering Service | Bounce | Google Groups | Human Verifier | Total Validated |
|---|---|---|---|---|
| Proofpoint | 114 | 11 | 11 | 136 |
| Barracuda | 51 | 0 | 1 | 52 |
| Cisco | 25 | 1 | 2 | 28 |
| Others | 111 | 0 | 0 | 111 |
| Total | 301 | 12 | 14 | 327 |

**Table 4: The number of domains validated using each technique and the filtering services those domains used.**

results agree with our inference results for both the email provider for the domain and their bypass configuration status.

The first technique takes advantage of bounce messages. For some organizations that use Exchange Online, we can send email to a non-existent RCPT address and the organization will send a bounce message in response that includes the delivery path of the original message [66]. For organizations where Exchange Online is misconfigured to allow bypass, the delivery path allows us to verify that the first server to receive the message is indeed an Exchange Online server (as expected when bypassing). This technique only applies to organizations that have Exchange Online misconfigured, do not enable Directory-Based Edge Blocking,[13] and generate bounce messages. Of the 955 organizations that misconfigure Exchange Server, 301 (32%) of them generate bounce messages, and in all cases they agree with our inference results.

The second technique uses responses from Google Groups administrative addresses to validate domains using Gmail. For example, a Google Group `group@univ.edu` always has a special address `group+unsubscribe@univ.edu` for unsubscribing. When sending email to the unsubscribe address from an account outside the group, Google Groups responds with an error that encloses the delivery path of the original message. We can verify that the first server on the delivery path is a Gmail server rather than the domain's filtering service. This approach, however, requires identifying Google Groups at organizations. Unfortunately, systematically searching for such addresses only discovered groups at 12 domains.

Our last technique involves personal contacts at the organization. We attempt to directly deliver email to our contacts email address. If the organization has securely configured their mail server, then it should reject our delivery attempt during the SMTP session. If the server is misconfigured, then delivery is successful and we ask our contact to forward the delivered message to us. We then verify that the server we used for delivery is indeed the first server on the delivery path (rather than the server of a filtering service).

## 5 ETHICS AND DISCLOSURE

There are two types of ethical considerations in our work that we discuss here: potential impacts to both humans and organizations.

A straightforward approach to a study such as ours would be to simply attempt to bypass the mail filtering services used by domains under test and then leverage widely-used mail content features (e.g., embedded links to images) to establish delivery. Indeed, in discussions with our IRB office, we have been informed that sending such unsolicited messages to individuals and evaluating if they are received would not be considered human subjects research, as we would not be collecting information *about* the person. However, we are sensitive that such emails *still* incur a de minimus nuisance cost on recipients (i.e., reading the message and choosing what to do in response) and thus our methodology has been carefully designed to focus on machine-to-machine communications whenever possible. However, in a minority of cases — when the domain is hosted by Exchange Online, is also configured to filter out invalid email addresses, and is incorrectly configured to allow its mail filtering service to be bypassed — we may deliver a single email to a role-based address (e.g., postmaster) whose identity is unknown to us.[14] In these cases, we solicit no response and perform disclosure by explaining the purpose of the study and then implications of receiving the email.[15] Further, when interviewing our institution's postmaster they confirmed that a single message would represent "a drop in the bucket" of the mail they receive on a daily basis and would not constitute a significant differential burden.

The second issue is that our work identifies vulnerable organizations whose mail configuration allows their mail filtering service to be bypassed. To avoid unnecessarily enabling malicious parties, we do not name vulnerable domains. We have disclosed these findings to the affected email filtering services (we explain this choice more fully in Appendix B). We interacted closely with three providers (Proofpoint, Forcepoint, HornetSecurity) who indicated they would notify their customers, and/or share best practice documentation with them.[16] Four vendors (Mimecast, Broadcom, CloudFlare and AppRiver) acknowledged our findings and two of these (Mimecast and Broadcom) indicated they would consider contacting their customers, and make improvements to their documentation. Five other vendors (Barracuda, Cisco, Sophos, Fortinet and N-able) consider the issue "out of scope" since it relates to their customer's configuration rather than a vulnerability in the service itself. Three others (TrendMicro, Trellix, Spamhero) have not responded to our outreach despite multiple attempts to contact them.

## 6 RELATED WORK

There is a large body of prior work focused on email security and infrastructure. We highlight here the publications and reports most closely related to our study. One popular line of research examines the deployment of different email security and encryption protocols. These include efforts to characterize the real-world deployment and challenges related to STARTTLS [14, 17, 23, 37, 50], SPF [11], DKIM, DMARC [8, 9, 12, 14, 17, 25, 33, 57, 62, 65], DANE [4, 30, 31], and PGP [59]. Separate from security pitfalls and solutions, prior work has also investigated email delivery and email service provisioning. Notably, Afergan et al. [1] examine the latency and loss aspects of email delivery and Holzbauer et al. [24] investigate protocol support in email delivery using passive DNS. Rijswijk et al [64] describe the growth of three email providers (as measured by MX records) and Liu et al. [34] provide a large-scale measurement documenting the change of email service provisioning over time.

[13] Of the 1,352 domains we inferred as using Microsoft Exchange, 775 (57%) of them have enabled Directory-Based Edge Blocking.

[14] This is similar to the approach taken by Bennett et al. in their 2022 IMC paper on inferring SPF vulnerabilities [8].

[15] We also provide a link that they can use to opt out of any future messages.

[16] We have witnessed associated changes to Proofpoint's documentation during this time.

The prior research most related to our work focuses on the effectiveness, deployment, and adoption of cloud-based email filtering services. This literature includes Rahmad et al.'s [54] comparative study of the effectiveness of different cloud-based email filtering services, industry reports on how to defeat Proofpoint's spam filtering [49], and Fiebig et al.'s [15] and Liu et al.'s [34] measurement studies on the adoption of cloud-based filtering services. Notably, while both of these two groups identify and document the increasing use of such filtering services, they do not investigate the security implications these changes. It is these implications that motivate our work to understand the integrity of mail filtering deployments.

## 7 DISCUSSION

The problems highlighted by this paper are superficially about a set of independent failures in administrative configuration. However, the underlying reasons for these failures all stem from the larger issue of architectural inadequacy. Email, like many legacy Internet services, was designed around a simple use case that is now out of step with modern demands. In the examples explored in this paper, the domain owner desired to reliably route inbound mail through an *ordered set* of cloud-hosted services: first to one third-party service (the filtering service) and *then*, after filtering, to another third-party service (the email provider). However, this desire cannot be expressed in the existing architecture for mail delivery. Instead, the domain owner's security depends on careful coordination between its own administrators, the email filtering service and the email provider to cobble together these semantics. Unsurprisingly, this ad hoc approach is rife with opportunities for failure.[17]

First, the separation of concerns is not naturally aligned with the interests and capabilities of the parties. The filtering service — the entity whose very existence motivates a bypass — is itself *incapable* of guaranteeing the integrity of mail delivery. It can forward filtered mail on to the hosting provider, but it cannot restrict from whom that provider accepts email. Only the hosting provider can implement such a restriction, but it may not always do so. Indeed, many hosting providers do not, and in this case there is no recourse for a domain owner (except to switch providers).

However, even if such a mechanism is available, it is only effective if a domain's administrator knows of its existence and importance. We note that the documentation for a number of filtering services makes no mention of the need to configure Gmail or Exchange Online to only accept mail from their servers. Even knowing that such a mechanism exists, email administrators must then implement such restrictions correctly. This can be difficult when, in one significant case, the filtering service's documentation for this step is inaccurate (in a way that ensures that bypass is possible). This knowledge issue must be overcome by each domain owner, even though the average email administrator is likely far less facile with email security than the staff at the cloud services being used.

Second, the complex federated nature of this service composition may impact (or at least be perceived to impact) email deliverability in a way that causes domain owners to favor "open" (i.e., non-secure) implementations. For example, at least one major mail filtering service warns in its documentation that enabling inbound mail restrictions on Gmail may lead to some email being dropped. Anecdotally, we observe a number of domains including Gmail's servers as "backup" entries in their MX records, presumably to tolerate a failure of the mail filtering service. However, this fault-tolerant configuration only works if Gmail is configured to accept mail from anyone (and hence, is bypassable). An added complication is that some configuration changes can take longer to propagate than others and there may be no mechanism to validate that such propagation completed. Accordingly, some email filtering tutorials strongly advise waiting 24–48 hours between updating MX records and applying the correct configuration on the email provider's side [60]. Such indeterminacy, coupled with the potential risks to deliverability, can cause email administrators to forego secure configurations to avoid service disruptions.

Finally, even in the best case, when everything is configured correctly, the integrity of the mail delivery path rests solely on the integrity of the source IP address, a design whose fragile security properties has long been understood [7]. Indeed, if an attacker can spoof the source address of a domain's email filtering service (e.g. communicate to Gmail servers as though one were Proofpoint) it is entirely likely that their email will avoid all filtering [6, 52].

Today's email filtering ecosystem repurposes the tools available from existing email and DNS protocols, designed long before widespread cloud deployments. The deficiencies of this approach, identified in this paper, highlight the need for a modern architecture for composing cloud services (such as email filtering) in a way that cleanly supports strong integrity, simple configuration and transparent auditability.

## 8 CONCLUSION

Organizations have increasingly turned to cloud-based email filtering services to defend against sophisticated email threats. These services filter by interposing between senders and an organization's email server. However, for this filtering function to be effective, organizations need to configure their email server to only accept email from their filtering service. Otherwise, malicious actors can bypass the filtering service by sending directly to the organization's email server. Using a range of `com` and `edu` domains as a case study, we empirically demonstrate that such bypasses are widely feasible: 80% of the domains are configured to allow such actions. Our work highlights the stresses placed on our legacy network architecture as it is asked to solve problems, such as securely composing cloud services, that were never part of its original design.

## ACKNOWLEDGMENTS

We thank the anonymous reviewers for their insightful and constructive feedback. Many thanks also to Cindy Moore and Jennifer Folkestad for operational support, to Liz Izhikevich and Gautam Akiwate for providing data access, and our various personal contacts for helping validate the results. Funding for this work was provided in part by NSF grant CNS-2152644, the Irwin Mark and Joan Klein Jacobs Chair in Information and Computer Science, the CSE Professorship in Internet Privacy and/or Internet Data Security, a generous gift from Google, and operational support from the UCSD Center for Networked Systems.

[17] While we discuss the challenge of enforcing an ordering among cloud providers, our takeaways generalize to other scenarios where incentives are not naturally aligned.

# REFERENCES


[1] Mike Afergan and Robert Beverly. 2005. The State of the Email Aaddress. *ACM SIGCOMM Computer Communication Review (CCR)* 35, 1 (2005), 29–36.

[2] Tony Akers. 2018. How attackers bypass third-party mail filtering to Office 365. (Nov. 2018). https://practical365.com/how-to-ensure-your-third-party-filtering-gateway-is-secure

[3] Manos Antonakakis, Tim April, Michael Bailey, Matt Bernhard, Elie Bursztein, Jaime Cochran, Zakir Durumeric, J Alex Halderman, Luca Invernizzi, Michalis Kallitsis, Deepak Kumar, Chaz Lever, Zane Ma, Joshua Mason, Damian Menscher, Chad Seaman, Nick Sullivan, Kurt Thomas, and Yi Zhou. 2017. Understanding the Mirai Botnet. In *Proceedings of the 26th USENIX Security Symposium*. Vancouver, BC, Canada, 1093–1110.

[4] Md. Ishtiaq Ashiq, Weitong Li, Tobias Fiebig, and Taejoong Chung. 2023. You've Got Report: Measurement and Security Implications of DMARC Reporting. In *Proceedings of the 32nd USENIX Security Symposium*. Anaheim, CA, USA, 4123–4137.

[5] Barracuda. 2023. Email Security Gateway. (Sept. 2023). https://www.barracuda.com/products/email-protection/email-security-gateway

[6] Barracuda. 2023. How to Configure Microsoft 365 for Inbound and Outbound Mail. (Feb. 2023). https://campus.barracuda.com/product/emailgatewaydefense/doc/96022752/step-2-configure-microsoft-365-for-inbound-and-outbound-mail/.

[7] Steven M. Bellovin. 1989. Security Problems in the TCP/IP Protocol Suite. *ACM SIGCOMM Computer Communication Review (CCR)* 19, 2 (1989), 32–48.

[8] Nathaniel Bennett, Rebekah Sowards, and Casey Deccio. 2022. Spfail: Discovering, Measuring, and Remediating Vulnerabilities in Email Sender Validation. In *Proceedings of the 22nd ACM Internet Measurement Conference (IMC)*. Nice, France, 633–646.

[9] Jianjun Chen, Vern Paxson, and Jian Jiang. 2020. Composition Kills: A Case Study of Email Sender Authentication. In *Proceedings of the 29th USENIX Security Symposium*. Virtual Event, 2183–2199.

[10] Cisco. 2022. Configure Microsoft 365 with Secure Email. (Dec. 2022). https://www.cisco.com/c/en/us/support/docs/security/cloud-email-security/214812-configuring-office-365-microsoft-with.html.

[11] Stefan Czybik, Micha Horlboge, and Konrad Rieck. 2023. Lazy Gatekeepers: A Large-Scale Study on SPF Configuration in the Wild. In *Proceedings of the 23rd ACM Internet Measurement Conference (IMC)*. Montreal, QC, Canada.

[12] Casey Deccio, Tarun Yadav, Nathaniel Bennett, Alden Hilton, Michael Howe, Tanner Norton, Jacob Rohde, Eunice Tan, and Bradley Taylor. 2021. Measuring Email Sender Validation in the Wild. In *Proceedings of the 17th International Conference on emerging Networking EXperiments and Technologies (CoNEXT)*. 230–242.

[13] Zakir Durumeric, David Adrian, Ariana Mirian, Michael Bailey, and J. Alex Halderman. 2015. A Search Engine Backed by Internet-Wide Scanning. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security (CCS)*. Denver, Colorado, USA, 542–553.

[14] Zakir Durumeric, David Adrian, Ariana Mirian, James Kasten, Elie Bursztein, Nicolas Lidzborski, Kurt Thomas, Vijay Eranti, Michael Bailey, and J. Alex Halderman. 2015. Neither Snow Nor Rain Nor MITM...: An Empirical Analysis of Email Delivery Security. In *Proceedings of the 2015 Internet Measurement Conference (IMC)*. Tokyo, Japan, 27–39.

[15] Tobias Fiebig, Seda Gurses, Carlos H. Ganan, Erna Kotkamp, Fernando Kuipers, and Taritha Sari. 2023. Heads in the Clouds? Measuring Universities' Migration to Public Clouds: Implications for Privacy & Academic Freedom. *Proceedings of the Privacy Enhancing Technologies Symposium (PETS)* 2 (2023), 117–150.

[16] Forrester. 2023. The Forrester Wave: Enterprise Email Security, Q2 2023. (June 2023). https://reprints2.forrester.com/#/assets/2/108/RES178496/report

[17] Ian D. Foster, Jon Larson, Max Masich, Alex C. Snoeren, Stefan Savage, and Kirill Levchenko. 2015. Security by Any Other Name: On the Effectiveness of Provider Based Email Security. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security (CCS)*. Denver, Colorado, USA, 450–464.

[18] Abhishek Ghosh. 2019. Fix: gsuite username is a reserved word Error (postmaster, abuse email). (May 2019). https:// thecustomizewindows.com/2019/05/fix-gsuite-username- is-a-reserved-word-error-postmaster-abuse-email/.

[19] Google. 2022. About CrUX. (June 2022). https://developer.chrome.com/docs/crux/about/

[20] Google. 2023. Handling reports of abuse and technical issues. (May 2023). https://support.google.com/a/answer/33389

[21] The Radicati Group. 2023. Secure Email — Market Quadrant 2023. (March 2023). https://docs.broadcom.com/doc/radicati-secure-email-market-quadrant-2023

[22] Justin Hoeft. 2021. Google Workspace Rejecting Sophos Setup Messages (and other important messages). (Jan. 2021). https://community.sophos.com/sophos-email/f/discussions/133526/google-workspace-rejecting-sophos-setup-messages-and-other-important-messages

[23] Ralph Holz, Johanna Amann, Olivier Mehani, Matthias Wachs, and Mohamed Ali Kaafar. 2015. TLS in the Wild: An Internet-wide Analysis of TLS-based Protocols for Electronic Communication. *arXiv preprint arXiv:1511.00341* (2015).

[24] Florian Holzbauer, Johanna Ullrich, Martina Lindorfer, and Tobias Fiebig. 2022. Not that Simple: Email Delivery in the 21st Century. In *2022 USENIX Annual Technical Conference (USENIX ATC)*. Carlsbad, CA, USA, 295–308.

[25] Hang Hu and Gang Wang. 2018. End-to-End Measurements of Email Spoofing Attacks. In *Proceedings of the 27th USENIX Security Symposium*. Baltimore, MD, 1095–1112.

[26] Liz Izhikevich, Gautam Akiwate, Briana Berger, Spencer Drakontaidis, Anna Ascheman, Paul Pearce, David Adrian, and Zakir Durumeric. 2022. ZDNS: A Fast DNS Toolkit for Internet Measurement. In *Proceedings of the 22nd ACM Internet Measurement Conference (IMC)*. Nice, France, 33–43.

[27] S. Kitterman. 2014. *Sender Policy Framework (SPF) for Authorizing Use of Domains in Email, Version 1*. RFC 7208. RFC Editor. http://www.rfc-editor.org/rfc/rfc7208.txt http://www.rfc-editor.org/rfc/rfc7208.txt.

[28] Dr. John C. Klensin. 2008. Simple Mail Transfer Protocol. RFC 5321. (Oct. 2008). https://doi.org/10.17487/RFC5321

[29] KnowBe4. 2023. How to Whitelist by IP Address in Google Workspace. (Oct. 2023). https://support.knowbe4.com/hc/en-us/articles/115002797527-Whitelisting-by-IP-Address-in-Google-Workspace.

[30] Hyeonmin Lee, Md. Ishtiaq Ashiq, Moritz Müller, Roland van Rijswijk-Deij, Taekyoung "Ted" Kwon, and Taejoong Chung. 2022. Under the Hood of DANE Mismanagement in SMTP. In *Proceedings of the 31st USENIX Security Symposium*. Boston, MA, USA, 1–16.

[31] Hyeonmin Lee, Aniketh Gireesh, Roland van Rijswijk-Deij, Taekyoung "Ted" Kwon, and Taejoong Chung. 2020. A Longitudinal and Comprehensive Study of the DANE Ecosystem in Email. In *Proceedings of the 29th USENIX Security Symposium*. Virtual Event, 613–630.

[32] Frank Li, Zakir Durumeric, Jakub Czyz, Mohammad Karami, Michael Bailey, Damon McCoy, Stefan Savage, and Vern Paxson. 2016. You've Got Vulnerability: Exploring Effective Vulnerability Notifications. In *Proceedings of the 25th USENIX Security Symposium*. Austin, TX, USA, 1033–1050.

[33] Enze Liu, Gautam Akiwate, Mattijs Jonker, Ariana Mirian, Grant Ho, Geoffrey M. Voelker, and Stefan Savage. 2023. Forward Pass: On the Security Implications of Email Forwarding Mechanism and Policy. In *Proceedings of the 8th IEEE European Symposium on Security and Privacy (EuroS&P)*. Delft, Netherlands.

[34] Enze Liu, Gautam Akiwate, Mattijs Jonker, Ariana Mirian, Stefan Savage, and Geoffrey M. Voelker. 2021. Who's Got Your Mail? Characterizing Mail Service Provider Usage. In *Proceedings of the 21st ACM Internet Measurement Conference (IMC)*. Virtual Event, 122–136.

[35] Yu Liu, Matthew R. Squires, Curtis R. Taylor, Robert J. Walls, and Craig A. Shue. 2019. Account Lockouts: Characterizing and Preventing Account Denial-of-Service Attacks. In *Security and Privacy in Communication Networks*, Songqing Chen, Kim-Kwang Raymond Choo, Xinwen Fu, Wenjing Lou, and Aziz Mohaisen (Eds.). Cham, 26–46.

[36] Zoho Mail. 2023. Spam Control Guidelines and Best Practices. (2023). https://www.zoho.com/mail/help/guidelines-spam-control.html

[37] Wilfried Mayer, Aaron Zauner, Martin Schmiedecker, and Markus Huber. 2016. No Need for Black Chambers: Testing TLS in the E-mail Ecosystem at Large. In *Proceedings of the 2016 International Conference on Availability, Reliability and Security (ARES)*. Salzburg, Austria, 10–20.

[38] Microsoft. 2022. External Domain Name System records for Office 365. (Dec. 2022). https://learn.microsoft.com/en-us/microsoft-365/enterprise/external-domain-name-system-records

[39] Microsoft. 2023. Configure mail flow using connectors in Exchange Online. (May 2023). https://learn.microsoft.com/en-us/exchange/mail-flow-best-practices/use-connectors-to-configure-mail-flow/use-connectors-to-configure-mail-flow

[40] Microsoft. 2023. How to set up a multifunction device or application to send email using Microsoft 365 or Office 365. (March 2023). https://learn.microsoft.com/en-us/exchange/mail-flow-best-practices/ how-to-set-up-a-multifunction-device-or-application-to-send- email-using-microsoft-365-or-office-365.

[41] Microsoft. 2023. Mail flow rules (transport rules) in Exchange Online. (Feb. 2023). https://learn.microsoft.com/en-us/exchange/security-and-compliance/mail-flow-rules/mail-flow-rules

[42] Microsoft. 2023. Manage mail flow using a third-party cloud service with Exchange Online. (Feb. 2023). https://learn.microsoft.com/en-us/exchange/mail-flow-best-practices/manage-mail-flow-using-third-party-cloud

[43] Microsoft. 2023. Office 365 URLs and IP address ranges. (April 2023). https://learn.microsoft.com/en-us/microsoft-365/enterprise/urls-and-ip-address-ranges?view=o365-worldwide

[44] Microsoft. 2023. Set up connectors for secure mail flow with a partner organization in Exchange Online. (Feb. 2023). https://learn.microsoft.com/en-us/exchange/mail-flow-best-practices/use-connectors-to-configure-mail-flow/set-up-connectors-for-secure-mail-flow-with-a-partner.

[45] Microsoft. 2023. Use Directory-Based Edge Blocking to reject messages sent to invalid recipients in Exchange Online. (Feb. 2023). https://learn.microsoft.com/en-us/exchange/mail-flow-best-practices/use-directory-based-edge-blocking.

[46] Mimecast. 2023. Email Security Cloud Gateway – Connect Process – Office 365 mail lockdown. (March 2023). https://community.mimecast.com/s/article/email-security-cloud-gateway-connect-process-0365-mail-lockdown.

[47] Mimecast. 2023. Email Security Cloud Gateway – Setting Up Your Inbound Email. (March 2023). https://community.mimecast.com/s/article/email-security-cloud-

gateway-setting-up-your-inbound-email.
[48] o365info. 2023. How to find Microsoft 365 MX record. (Aug. 2023). https://o365info.com/microsoft-365-mx-record/
[49] Will Pearce and Nick Landers. 2019. The answer to life, the universe, and everything offensive security. (Sept. 2019). https://github.com/moohax/Talks/blob/master/slides/DerbyCon19.pdf
[50] Damian Poddebniak, Fabian Ising, Hanno Böck, and Sebastian Schinzel. 2021. Why TLS is Better Without STARTTLS: A Security Analysis of STARTTLS in the Email Context. In *Proceedings of the 30th USENIX Security Symposium*. Virtual Event, 4365–4382.
[51] Proofpoint. 2023. Configuring Google Workspace (Gsuite) for Proofpoint Essentials. (May 2023). https://help.proofpoint.com/Proofpoint_Essentials/Email_Security/Administrator_Topics/hostedemailservices/Configuring_Google_Workspace_(Gsuite)_for_Proofpoint_Essentials
[52] Proofpoint. 2023. Configuring Microsoft 365 for Proofpoint Essentials. (March 2023). https://web.archive.org/web/20230328135954/https://help.proofpoint.com/Proofpoint_Essentials/Email_Security/Administrator_Topics/hostedemailservices/Configuring_Microsoft_365_for_Proofpoint_Essentials
[53] Proofpoint. 2023. Email Protection Solutions - Secure Email Provider. (Sept. 2023). https://www.proofpoint.com/us/products/email-security-and-protection/email-protection
[54] F Rahmad, Y Suryanto, and K Ramli. 2020. Performance Comparison of Anti-Spam Technology Using Confusion Matrix Classification. In *IOP Conference Series: Materials Science and Engineering*, Vol. 879. 012076.
[55] Keegan Ryan, Kaiwen He, George Arnold Sullivan, and Nadia Heninger. 2023. Passive SSH Key Compromise via Lattices. In *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security (CCS)*. Copenhagen, Denmark, 2886–2900.
[56] Jamie Scaife. 2023. Using SPF Macros to Solve the Operational Challenges of SPF. (10 2023). https://www.jamieweb.net/blog/using-spf-macros-to-solve-the-operational-challenges-of-spf/
[57] Kaiwen Shen, Chuhan Wang, Minglei Guo, Xiaofeng Zheng, Chaoyi Lu, Baojun Liu, Yuxuan Zhao, Shuang Hao, Haixin Duan, Qingfeng Pan, and Min Yang. 2021. Weak Links in Authentication Chains: A Large-scale Analysis of Email Sender Spoofing Attacks. In *Proceedings of the 30th USENIX Security Symposium*. Virtual Event, 3201–3217.
[58] Sophos. 2023. Configure Microsoft 365. (March 2023). https://docs.sophos.com/central/customer/help/en-us/ManageYourProducts/EmailSecurity/SophosGateway/ExternalServices/ConfigureM365/index.html
[59] Christian Stransky, Oliver Wiese, Volker Roth, Yasemin Acar, and Sascha Fahl. 2022. 27 Years and 81 Million Opportunities Later: Investigating the Use of Email Encryption for an Entire University. In *Proceedings of the 2022 IEEE Symposium on Security and Privacy*. 860–875.
[60] Vircom Support. 2023. Configuring Google Workspace (Gsuite) for Proofpoint Essentials. (April 2023). https://vircomhelp.freshdesk.com/support/solutions/articles/48001171784-configuring-google-workspace-gsuite-for-proofpoint-essentials.
[61] Ali Tajran. 2023. How to configure Microsoft 365 to only accept mail from third-party spam filter. (May 2023). https://www.alitajran.com/only-accept-from-third-party-spam-filter.
[62] Dennis Tatang, Florian Zettl, and Thorsten Holz. 2021. The Evolution of DNS-based Email Authentication: Measuring Adoption and Finding Flaws. In *Proceedings of the 2021 International Symposium on Research in Attacks, Intrusions and Defenses (RAID)*. San Sebastian, Spain, 354–369.
[63] TrendMicro. 2023. Adding Office 365 Inbound Connectors. (Oct. 2023). https://success.trendmicro.com/dcx/s/solution/000250836-trend-micro-email-security-integration-with-microsoft-office-365.
[64] Roland van Rijswijk-Deij, Mattijs Jonker, Anna Sperotto, and Aiko Pras. 2016. A High-Performance, Scalable Infrastructure for Large-Scale Active DNS Measurements. *IEEE Journal on Selected Areas in Communications* 34, 6 (2016), 1877–1888.
[65] Chuhan Wang, Kaiwen Shen, Minglei Guo, Yuxuan Zhao, Mingming Zhang, Jianjun Chen, Baojun Liu, Xiaofeng Zheng, Haixin Duan, Yanzhong Lin, and Qingfeng Pan. 2022. A Large-scale and Longitudinal Measurement Study of DKIM Deployment. In *Proceedings of the 31st USENIX Security Symposium*. Boston, MA, USA, 1185–1201.
[66] Wikipedia. 2023. Bounce message. (May 2023). https://en.wikipedia.org/wiki/Bounce_message
[67] Sophia Willows. 2023. How to access your Google Workplace's postmaster@ inbox. (May 2023). https://sophiabits.com/blog/how-to-access-google-postmaster-email

# A HANDLING SPF EDGE CASES

This section provides additional information on how we handle two edge cases: SPF records that use the `include` mechanism and SPF macros.

For SPF records that use the `include` mechanism, we expand them recursively to a maximum depth of three. At each recursion, we first check for the presence of SPF records for each provider (e.g., `include:_spf.google.com` for Gmail). If we find no match, we proceed to check if any IP addresses (`ip4` record) belong to the suite of outbound IP address used by each provider (e.g., Exchange Online's list of outbound IP address [43]). If either of the two checks succeeds, we label the domain as using the corresponding provider.

Besides the `include` mechanism, we also handle SPF records that contain macros, which are used by 8% (120/1,601) of the domains in our dataset. SPF macros provide a mechanism for dynamic SPF policies. Namely, instead specifying a list of IP addresses, it defines special sequences that are interpreted at runtime by the receiving Mail Transfer Agent (MTA). For example, the macro `%{i}` expands to the sender's IP address. If a domain's SPF record is "`v=spf1 include:%{i}.spf.domain.com -all`", the receiving MTA will replace `%{i}` with the sender's IP address and then perform the SPF check by sending a DNS TXT query to `<sender's_IP_address>.spf.domain.com`. Another common macro is `%{d}`, which expands to the domain name of the sender's email address. SPF macros are designed to help avoid the ten lookup limit imposed by the DNS protocol and enable more dynamic SPF policies [56].

For a domain that has an SPF record with macros, we determine if it allows an email hosting provider to send on its behalf as if we received an email from that email hosting provider. Specifically, we expand the macros by replacing `%{i}` with an outbound IP address used by the email hosting provider and `%{d}` with the target domain's name. We then issue the DNS TXT query to the target domain. If the response indicates that the IP address used is allowed, we label the domain as using the email service provider.

# B CHOICE OF DISCLOSURE

This section presents our reasoning for disclosing to the vendors of the email filtering services, as opposed to individualized disclosure to each of the domain owners themselves. The first issue is simply the limitation of scale. Identifying the appropriate contacts at each of over 1,200 organizations, contacting them, and then responding to their follow-up requests, exceeds our resources as a small university research group. Indeed, this reality is why aggregate disclosure to service providers has long been the norm in the research community for studies of this type (e.g., similar studies of mail authentication vulnerabilities [9, 33], of software vulnerability prevalence [55], botnet compromise [3], etc). The second issue is that even where such contact is feasible, prior research has consistently demonstrated that doing so has limited effectiveness [8, 32]. Indeed, Bennett et al. reported that over 80% of the domains contacted were unresponsive [8]. An unsolicited contact from an external party with no existing relationship with an organization is commonly untrusted and rarely prioritized. Instead, we elected to work directly and closely with filtering service providers to update their documentation, notify their customers (with whom they *do* have an existing business relationship) and resolve the issues identified (Section 5).