# Forward Pass: On the Security Implications of Email Forwarding Mechanism and Policy


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

Gautam Akiwate
*Stanford University*
*Stanford, CA, USA*
*gakiwate@cs.stanford.edu*

Mattijs Jonker
*University of Twente*
*Enschede, Netherlands*
*m.jonker@utwente.nl*

Ariana Mirian
*UC San Diego*
*La Jolla, CA, USA*
*amirian@cs.ucsd.edu*

Grant Ho
*UC San Diego*
*La Jolla, CA, USA*
*grho@eng.ucsd.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*—The critical role played by email has led to a range of extension protocols (e.g., SPF, DKIM, DMARC) designed to protect against the spoofing of email sender domains. These protocols are complex as is, but are further complicated by automated email forwarding — used by individual users to manage multiple accounts and by mailing lists to redistribute messages. In this paper, we explore how such email forwarding and its implementations can break the implicit assumptions in widely deployed anti-spoofing protocols. Using large-scale empirical measurements of 20 email forwarding services (16 leading email providers and four popular mailing list services), we identify a range of security issues rooted in forwarding behavior and show how they can be combined to reliably evade existing anti-spoofing controls. We further show how these issues allow attackers to not only deliver spoofed email messages to prominent email providers (e.g., Gmail, Microsoft Outlook, and Zoho), but also reliably spoof email on behalf of tens of thousands of popular domains including sensitive domains used by organizations in government (e.g., `state.gov`), finance (e.g., `transunion.com`), law (e.g., `perkinscoie.com`) and news (e.g., `washingtonpost.com`) among others.**


## 1. Introduction

Email has long been a uniquely popular medium for social engineering attacks.[1] While it is widely used for both unsolicited business correspondence as well as person-to-person communications, email provides no intrinsic integrity guarantees. In particular, the baseline SMTP protocol provides no mechanism to establish if the purported sender of an email message (e.g., From: `Anthony.Blinken@state.gov`) is in fact genuine.

To help address this issue, starting in the early 2000's, the email operations community introduced multiple anti-spoofing protocols, including the Sender Policy Framework (SPF) [2], DomainKeys Identified Mail (DKIM) [3] and Domain-based Message Authentication Reporting and Conformance (DMARC) [4], each designed to tighten controls on which parties can successfully deliver mail purporting to originate from particular domain names. However, these protocols had the disadvantage of being both post-hoc (needing to support existing email deployments and conventions) and piecemeal (each addressing slightly different threats in slightly different ways). As a result, the composition of these protocols is complex and hard to reason about, leading to a structure that Chen et al. recently demonstrated can enable a range of evasion attacks [5].

In this paper, we explore the unique aspects of this problem created as a result of *email forwarding*, which is commonly used by both individuals (i.e., to aggregate mail from multiple accounts) and organizations (i.e., for mailing list distribution). While clearly useful, forwarding introduces a range of new interaction complexities. First, forwarding involves three parties instead of two (the sender, the forwarder, and the receiver), where the "authenticity" of an email message is commonly determined by the party with the weakest security settings. Second, the intrinsic nature of email forwarding is to transparently send an existing message to a new address "on behalf" of its original recipient — a goal very much at odds with the anti-spoofing function of protocols such as SPF and DMARC. For this reason, forwarded email messages can receive special treatment based on various assumptions about how forwarding is used in practice. Finally, there is no single standard implementation of email forwarding. Different providers make different choices and the email ecosystem is forced to accommodate them. Unfortunately, some problematic implementation choices (e.g., permitting "open forwarding") incur no security impact on the implementing party but can jeopardize the security of downstream recipients. This inversion of incentives and capabilities creates additional challenges to mitigating forwarding vulnerabilities.

To characterize the nature of these issues, we conduct a large-scale empirical measurement study to infer and characterize the mail forwarding behaviors of 16 leading email providers and four popular mailing list services. From

1. In the 2021 Verizon Data Breach Investigation Report, phishing is implicated in 36% of the more than 4,000 data breaches investigated; and email-based attacks, including Business Email Compromise (BEC), completely dominate the social engineering attack vector [1].

these results, we identify a range of implicit assumptions and vulnerable features in the configuration of senders, receivers, and forwarders. Using a combination of these factors, we then demonstrate a series of distinct evasion attacks that bypass existing anti-spoofing protocols and allow the successful delivery of email with spoofed sender addresses (e.g., From: `Anthony.Blinken@state.gov`). These attacks affect both leading online email service providers (e.g., Gmail, Microsoft Outlook, iCloud, and Zoho) and mailing list providers/software (e.g., Google Groups and Gaggle). Moreover, some of these issues have extremely broad impact — affecting the integrity of email sent from tens of thousands of domains, including those representing organizations in the US government (spanning the majority of US cabinet domains, such as `state.gov` and `doe.gov`, as well as the domains of security agencies such as `odni.gov`, `cisa.gov`, and `secretservice.gov`), financial services (e.g., `transunion.com`, `mastercard.com`, and `discover.com`), news (e.g., `washingtonpost.com`, `latimes.com`, `apnews.com`, and `afp.com`), commerce (e.g., `unilever.com`, `dow.com`), and law (e.g., `perkinscoie.com`). Finally, in addition to disclosing these issues to their respective providers, we discuss the complexities involved in identifying, mitigating, and fixing such problems going forward.

## 2. Background

In this section, we describe the anatomy of a simple email transmission and the protocols used to authenticate such an email. We also present a high-level overview of how forwarding modifies the email delivery flow as a basis for a detailed description of different forwarding approaches and implementations in Section 3. Finally, we briefly survey related work on email security, particularly those whose insights we have built upon.

### 2.1. Simple Mail Transfer Protocol

The Simple Mail Transfer Protocol (SMTP) governs the addressing and delivery of Internet email [6]. Designed to mimic physical mail, SMTP specifies two distinct sets of headers that declare the sender and recipient(s) of an email message. An outer set of headers, the *SMTP Envelope Headers* (MAIL FROM and RCPT TO), tell email servers how to route and deliver email. In particular, the RCPT TO header identifies the message's recipient and the MAIL FROM header identifies where to send replies and bounce messages. An inner set of headers, the *Message Headers* (FROM and TO), are contained in the body of the SMTP message [7]. These correspond to the human-readable names and addresses set by email clients when the sending user creates an email message. These headers are strictly intended for human user-interface purposes (i.e., for populating the "To:" and "From:" fields in email clients) and they are not used for email routing. Figure 1 illustrates an example message with both sets of headers. Note that, although the addresses in the Envelope and Message headers frequently match (as they do in our example), they are not required to do so and there are both benign (e.g., email forwarding) and malicious (e.g., phishing) reasons for producing mismatched headers


Figure 1: Example SMTP headers in a transmission (inspired by Figure 3 in Chen et al. [5]).

(e.g., where the MAIL FROM address does not match the FROM address).

### 2.2. Email Spoofing Protections

The original SMTP design lacks authentication, which has made email spoofing attacks both possible and common. To mitigate these attacks, the community has proposed multiple mechanisms that focus on authenticating the *domain name* used by the purported sender.[2] Of these mechanisms, we focus on SPF [2], DKIM [3], and DMARC [4] given their wide adoption.

**Sender Policy Framework (SPF)** defines a list of IP addresses permitted to send email on behalf of a domain and a set of actions the recipient should take if they receive an email from an unauthorized IP address.[3] Domain owners specify this policy by publishing it in a DNS TXT record. Upon receiving an email message, the receiver fetches the list of authorized sender IP addresses by querying the domain in the email's MAIL FROM header. The recipient then verifies if the IP address of the sending server is included that list. If the verification fails, the receiver enforces the action (e.g., marking the email as spam) specified by the MAIL FROM domain in their SPF policy.

**DomainKeys Identified Mail (DKIM)** cryptographically binds an email message with its sending email domain. With DKIM, the sender signs an email (or certain elements of an email) and attaches a digital signature via a DKIM-SIGNATURE message header for future verification. Receivers later retrieve the signer's public key (in the form of a DNS TXT record) from the domain specified in the DKIM-SIGNATURE header and authenticate an email's signature using that key.

Sadly, neither SPF nor DKIM verify that an email's purported sender (i.e., the FROM header) truly wrote and sent it [5]. For example, an attacker could bypass DKIM by spoofing an email's FROM header, but then sign and attach a DKIM signature that uses key pairs linked to their own domain (since DKIM does not compare the signature's domain against the FROM domain). Attacks

2. True per-sender authentication has long floundered due to the lack of effective mechanisms for binding user identities with cryptographic credentials at scale. The best known protocol in this space, PGP, has been riddled with security and usability issues and remains, at best, a niche protocol. In this paper, we focus exclusively on domain-level sender authentication.

3. In addition to lists of raw IP addresses, SPF records can also "include" other SPF records by reference.

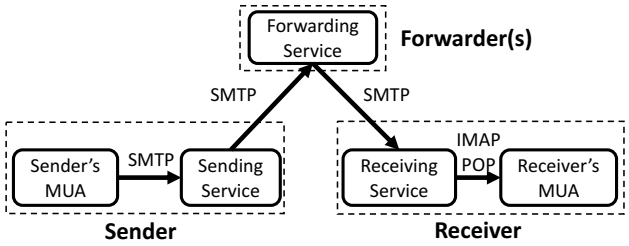


Figure 2: Email flow involving forwarding.

that exploit this lack of FROM header authentication motivated the creation of DMARC.

**Domain Message Authentication, Reporting, and Conformance (DMARC)** combines and extends SPF and DKIM to mitigate these security issues. Under DMARC, an email's receiver performs an "alignment test": checking if the domain in the FROM header matches the domain name verified by either SPF (the domain in the MAIL FROM header) or DKIM (the domain in the DKIM-SIGNATURE header). By default ("relaxed mode"), the alignment test only requires that the registered domains in the headers match (i.e., not the fully qualified domain name (FQDN)). However, domain owners can specify that recipients should follow the strict mode of the alignment test, which requires the FROM header's FQDN to exactly match the domain authenticated by SPF or DKIM.[4]

If the email passes either SPF or DKIM authentication, and the alignment test also passes, then DMARC considers the email authenticated. Otherwise, the receiver should implement the DMARC policy designated by the domain in the FROM header, selected from one of three options: NONE, QUARANTINE, or REJECT. A policy of NONE specifies that an email should be delivered as normal (and thus is often used for monitoring purposes [8], [9]), and REJECT specifies that the recipient mail server should drop the email without delivering it to the user. The QUARANTINE policy is not strictly defined (indicating only that the message should be treated "as suspicious") and allows each email provider considerable latitude in their implementation (e.g., setting a UI indicator or placing the email in a designated spam folder) [4].

## 2.3. Email Forwarding

Forwarding is ubiquitous in the email ecosystem and is necessitated by the wide use of mailing lists [10], email filtering services such as ProofPoint [11], and auto-forwarding employed by individual users for account aggregation [12], among others. As shown in Figure 2, forwarding alters the standard transmission flow of an email message. Instead of a direct transmission from the sender to the recipient, forwarding relays an email from the sender to an intermediate server and/or account, which then transmits a copy of the email to the final recipient. For simplicity we show a single forwarder in our example, but email can pass through multiple forwarders in common use cases.

Like normal receivers in direct mail transfer, forwarders are responsible for performing standard authentication checks on each email they receive. However, after authenticating a message, a forwarder often makes *changes* to the email headers and/or the email body based on the service it provides. The forwarder then sends the modified message to the final receiver (or next forwarder), which also performs authentication checks upon receiving the email. Finally, when a recipient receives and opens an email, the receiver's user agent (MUA) parses and displays the message to the user.

4. DMARC policy records are also stored as DNS TXT records.

## 2.4. Related Work

Email security has been a long-standing problem and a variety of prior research efforts have examined different aspects of it. One line of work focuses on understanding and defending against phishing attacks. This includes papers that design new tools for detecting both traditional phishing and sophisticated spearphishing attacks [13], [14], [15], [16], [17], [18], [19], [20], [21], [22], [23], [24], study the characteristics of real-world phishing attacks [25], [26], [27], [28], and examine the human aspect of such attacks [29], [30], [13], [31], [32], [33], [34], [35].

Another body of work investigates the security and deployment of email encryption mechanisms, such as PGP [36], [37], [38], [39], [40], DANE [41], [42], and STARTTLS [43], [44], [45], [46], [47].

A third research direction analyzes the security and deployment of anti-spoofing protocols such as SPF, DKIM and DMARC, with efforts from both industry and academia. The blogposts by Ullrich [48] and Haddouche [49] investigated approaches for bypassing DKIM and DMARC using malformed email messages. Other work has empirically measured the efficacy and deployment status of SPF, DKIM, and DMARC [50], [43], [44], [51], [52], [53], [54], as well as qualitatively characterized the factors that drive DMARC policy decisions [55].

The work most related to our own includes Chen et al.'s analysis of the security vulnerabilities introduced by protocol composition in modern email delivery [5], Shen et al.'s analysis [56] of modern sender spoofing attacks, and Wang et al.'s [57] analysis of email security under the experimental Authenticated Received Chain (ARC) protocol [58]. Of these, Chen et al. [5] do not consider forwarding at all and Wang et al. [57] focus on ARC and only consider one specific forwarding implementation as well (REM+MOD in Section 3), leaving many other vulnerable forwarding mechanisms and features unexplored.

Shen et al.'s work [56] is the closest in that it also examines open forwarding, but because they only consider one forwarding mechanism (what we label as REM in Section 3), they do not identify the significant scope of this issue. We build on and generalize this work to show, among other attacks, that attackers are able to practically abuse open forwarding to spoof *any* domain that includes the forwarding domain's SPF record in their own SPF record (a common practice when hosting email via Microsoft's Outlook service for example).

In summary, our paper builds on the insights of prior efforts, but focuses exclusively and deeply on the particular security challenges introduced by the design and features of common forwarding mechanisms, and their complex interactions with existing email protocols. Through systematic measurements and analysis, we not only show that prior work largely underestimates the risks of open

forwarding, but also reveal new attacks not discovered in prior work.

## 3. Email Forwarding in Practice

Despite the ubiquity of email forwarding, there is no single and universally agreed-upon method for how email services should implement forwarding, resulting in several different approaches [59]. This heterogeneity stems in part from the difficulty of balancing compatibility with anti-spoofing protocols and the functional goals of many forwarding use cases: to transparently hide the intermediate forwarder and present the illusion that the recipient receives the email directly from its original sender.

Absent a clear standard to depend on, we have used empirical measurements to *infer* the forwarding behavior deployed by prominent email providers and mailing list services. For each service, we created multiple test accounts, used them to forward email to recipient accounts we controlled, and then analyzed the resulting email headers to identify the forwarding mechanism employed. (Section 4.1 has a more detailed description of our methodology.)

We constructed a comprehensive and representative set of forwarding services by building on top of prior literature. In total, we studied 20 distinct, leading email forwarding services. We started by collecting all email providers studied in prior literature [5], [56], [50], [57]. We considered an email provider *out of scope* if it meets any of these four criteria: (a) it is no longer active (e.g., excite.com); (b) it does not accept US customers (e.g., all Chinese providers studied in prior work); (c) it is not open to public registration (e.g., cock.li); or (d) it does not support forwarding (e.g., Protonmail). Using these criteria, we identified 23 email providers. Next, we excluded five email providers that prohibited bulk registration (which prevents us from running large-scale measurements), leaving us with 18 email providers. We then identified and removed duplicate providers that are operated by the same vendor under different names, leading to a total of 14 distinct email providers. Finally, we augmented this set of forwarding services by searching for popular email providers that supported forwarding and widely-used mailing list services (a common use case overlooked in prior literature), adding two additional email providers (Mail2World and GoDaddy) and four mailing lists.

Our selection of email forwarding services covers a diverse set of countries and real-world use cases (personal and business email), and represents services used by the general public (used by over 46% of popular Alexa domains and government domains according to Liu et al. [60] ignoring email filtering services). We list all email providers and mailing lists in Table 1.

Through our measurements, we confirmed the use of three common approaches that are generally known through public documentation, and identified a fourth uncommon implementation used by Microsoft Outlook (hence referred to as Outlook) and Freemail.hu (hence referred to as Freemail). As summarized in Figure 3, in each approach the forwarder modifies the sender and recipient fields in the SMTP Envelope and Message headers before relaying the email to its recipient.

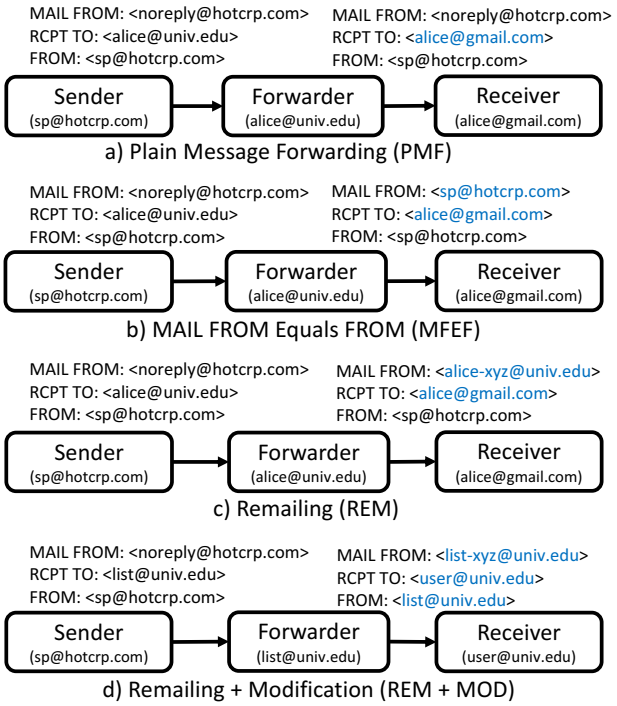


Figure 3: Four prevalent approaches to email forwarding. Addresses in blue correspond to header values rewritten during the forwarding process.

We now describe each of these approaches in detail using two running examples of common email forwarding use cases. In the first case, Alice has configured her university account (`alice@univ.edu`) to forward to her primary personal account (`alice@gmail.com`). When her university account receives email (e.g., from `sp@hotcrp.com`), forwarding retransmits it to `alice@gmail.com` in a way that makes it seem like the email comes directly from the sender (`sp@hotcrp.com`), rather than from her university account. In the second case, Bob sends an email to a mailing list (`list@univ.edu`), which redistributes (forwards) the email to the list's members (e.g., `user@univ.edu`).

**Plain Message-Forwarding (PMF):** Initially designed for the purpose of "source-routing" [61], PMF was one of the first forwarding mechanisms in wide use. Forwarders that use PMF only change the RCPT TO header from the forwarder's email account (`alice@univ.edu`) to the final recipient's address (`alice@gmail.com`), and leave all other fields untouched, as illustrated in Figure 3a. This approach achieves the goal of transparent forwarding. Changing the RCPT TO header will tell mail servers to send the email to the new address's account, and leaving the FROM header intact will cause the recipient's email client to display the initial sender (`sp@hotcrp.com`), rather than presenting `alice@univ.edu` as the sender.

**MAIL FROM Equals FROM (MFEF):** Similar to PMF, MFEF (Figure 3b) aims to achieve transparent forwarding by preserving the original sender's identity in the FROM header. Unlike the other forwarding approaches described in this section, MFEF is a custom forwarding implementation that appears to be used only by Outlook and

Freemail. A MFEF forwarder not only rewrites the RCPT TO header to the final recipient (alice@gmail.com), but it *also* sets the MAIL FROM header to be the same as the FROM header (from noreply@hotcrp.com to sp@hotcrp.com).

Email forwarded using PMF and MFEF often break SPF validation because the MAIL FROM domain typically does not list the forwarding server's IP address in its SPF allowlist; in our example, hotcrp.com does not list the email servers for univ.edu in its SPF allowlist. This incompatibility has hindered the adoption of SPF and DMARC [55], leading to provider-specific defenses and new anti-spoofing protocols that we describe in Section 4.

**Remailing (REM):** Unlike PMF and MFEF, remailing (aka redistribution) works well with SPF because this approach alters the headers in a way that resembles the action of the forwarder submitting a new message [62]. As shown in Figure 3c, the REM forwarder (univ.edu's mail server) first changes the RCPT TO header to specify the final recipient (alice@gmail.com). Additionally, the forwarder rewrites the MAIL FROM header so that it corresponds to an address in the forwarder's own domain (e.g., alice-xyz@univ.edu).[5]

However, even though REM interoperates with SPF, it can still fail DMARC authentication. Absent a valid DKIM header, email messages forwarded via REM will fail DMARC's alignment test because the FROM domain will not match the SPF-verified MAIL FROM domain.[6] This incompatibility has led to the common adoption of weaker DMARC policies, such as NONE and QUARANTINE instead of REJECT [55].

**Remailing with Modification (REM + MOD):** The final forwarding approach, Remailing with Modification (REM + MOD) [64], resolves these compatibility issues by sacrificing the goal of transparent forwarding. Email forwarded using REM + MOD will pass both SPF and DMARC. However, email forwarded with this approach will display the *forwarder* as the email's sender to the final recipient (hiding the identity of the original sender). Because of this functional change, most major email platforms do not adopt this approach, and it is used primarily by mailing list services such as Gaggle.

As shown in Figure 3d, with REM + MOD the forwarder modifies the headers just like it would during REM forwarding: changing the RCPT TO header to the final recipient (user@univ.edu) and the MAIL FROM header to an address in the forwarder's domain. Additionally, the forwarder rewrites the FROM header to match its account or an email address within its domain (e.g., list@univ.edu).

Although this forwarding approach produces email messages compatible with DMARC, we found that it also introduces a new set of security concerns and spoofing attacks (§ 5.4). At a high-level, because REM + MOD rewrites a forwarded email's headers to always pass SPF and DMARC checks, it enables an attacker to launder a spoofed email through a vulnerable forwarder such that it appears like a legitimate email message to the recipient.

5. The Sender Rewriting Scheme (RFC 5231 [63])provides a generic framework for how forwarders should rewrite the MAIL FROM header. However, email providers do not strictly follow this scheme and the exact email address after rewriting varies by implementation.

6. Many domains still do not implement DKIM for outbound email, and even those that do can have their user's DKIM signatures invalidated by mailing list software that adds content to a user's post [64].

| Email Provider | Forwarding Mechanism | Mailing List Service | Forwarding Mechanism |
|---|---|---|---|
| Fastmail | PMF | Gaggle | REM+MOD |
| Freemail.hu | MFEF | Google Groups | REM |
| GMX/Mail.com | REM | Mailman | REM |
| Gmail | REM | Listserv | REM |
| GoDaddy | REM | | |
| Hushmail | PMF | | |
| iCloud | PMF | | |
| Inbox.lv | REM | | |
| Mail.ru | PMF | | |
| Mail2World | PMF | | |
| Onet.pl/Op.pl | REM | | |
| Outlook/Hotmail/O365 | MFEF | | |
| Pobox | REM | | |
| Runbox | PMF | | |
| Yahoo | PMF | | |
| Zoho | REM | | |

TABLE 1: The providers and mailing list services we tested and the forwarding mechanisms they use. For providers that are operated by the same vendor under different names (e.g., GMX and Mail.com), we merge them into one row. O365 stands for Office 365.

Table 1 summarizes the default forwarding approach used by each of the email providers and mailing lists in our study.[7] The most common forwarding approach is remailing forwarding (REM), used by seven email providers (GMX, Gmail, GoDaddy, Inbox, Onet, Pobox, and Zoho) and three mailing lists (Google Groups, Listserv, and Mailman). Seven email providers, Fastmail, Hushmail, iCloud, Mail.ru, Mail2World, Runbox, and Yahoo, use plain-message forwarding (PMF). Outlook and Freemail use their own custom forwarding mechanism (MFEF) and, as described, Gaggle uses remailing with modification (REM+MOD) forwarding.

## 4. Assumptions and Vulnerable Features

In this section, we describe a range of email design and implementation weaknesses that lead to forwarding vulnerabilities. We start by exploring four assumptions made by anti-spoofing mechanisms that email forwarding can bypass and violate. We then examine three vulnerable forwarding features in the major forwarding approaches.

In each of these cases, we use active measurements — either of mail services themselves or the DMARC policies as stored in DNS — to document the prevalence of each issue among prominent domains and providers (summarized in Table 2). In the remainder of this section, we discuss the measurement methodology used to investigate and identify these issues and describe each vulnerability in turn. In the next section, we then show how these vulnerabilities can be combined to create complete and effective spoofing attacks involving a broad array of popular and sensitive domains.

7. A provider might forward differently when forwarding between internal accounts, and a mailing list might switch to a different forwarding mechanism to avoid issues caused by forwarding email messages from domains with stricter DMARC policies [65]. We do not consider these two cases.

| | Security Assumption or Feature | Implementation Aspect | Prevalence |
|---|---|---|---|
| § 4.2.1 | Domain will use actionable DMARC policies | DMARC None | Two-thirds of Alexa Top 1M |
| § 4.2.2 | Each domain uses its own infrastructure | Shared SPF record | All providers |
| § 4.2.3 | Quarantining is sufficient | Quarantine instead of reject | Outlook, Fastmail, GMX, Inbox.lv, Pobox |
| § 4.2.4 | Per-user DMARC overrides are fate-sharing | Domain whitelisting | All providers |
| § 4.3.1 | Users only forward to accounts they control | Open forwarding | Ten providers including Outlook and Fastmail |
| § 4.3.2 | Forwarded email from large providers benign | Relaxed validation | Gmail, Outlook, Mail.ru |
| § 4.3.3 | Adding DKIM signature increases deliverability | Unsolicited DKIM signatures | iCloud, Runbox, Hushmail |

TABLE 2: Summary of vulnerable security assumptions and forwarding features, the aspect of their implementation that leads to the vulnerability, and the prevalence of the vulnerability.

## 4.1. Methodology

For our experiments we created test *forwarding* accounts on all 20 forwarding services, test *recipient* accounts on all 16 major email providers, and mail servers for domains we control as the *sending* accounts. For Google Groups and Gaggle, we created mailing lists under our university's existing service and at `gaggle.email`, respectively. The other two mailing list services (Listserv and Mailman) rely upon a third-party backend mail server; we used Postfix [66] as the backend with DMARC enforced. We then created mailing lists under new domains we acquired for testing (e.g., `list@listserv.ourdomain.com`).

For each combination of forwarding and recipient accounts, we sent email using three different control domains in the FROM headers, each with the same SPF configuration but with distinct DMARC policies: NONE, QUARANTINE, and REJECT. Some services (e.g., Gmail and Outlook) will mark email messages sent from new domains as spam until there is sufficient user interaction with those messages. To avoid this startup effect, we "warmed up" our domains using a series of legitimate exchanges. In particular, from each domain, we sent legitimate (i.e., unspoofed) email that passed SPF, DKIM and DMARC to our accounts at each provider. Any message that was delivered to the spam folder we manually marked as "not spam". After this warm up period, we validated that legitimate (i.e., unspoofed) email from our domains was properly delivered to account inboxes in all cases.

Having primed our accounts, we assessed the prevalence of each vulnerability by sending legitimate and spoofed email messages to all pairwise combinations of our forwarding and recipient accounts.[8] We analyzed the headers and outcomes of these attempts, and recorded which parties exhibited vulnerable behavior. In particular, we configured all forwarders to forward email messages to all receivers and recorded whether each message was delivered to the inbox, spam folder, or rejected without delivery by each receiver. We also noted whether any UI warning was shown in the native web-based MUA.

## 4.2. Email Security Assumptions

Anti-spoofing mechanisms define a set of validation procedures which both explicitly and implicitly rely on assumptions about the behavior of domain holders, email providers and users. Here we identify four such assumptions that are crucial to these defenses in the direct single-hop delivery context, but do not necessarily hold in the presence of email forwarding.

**4.2.1. Domains use actionable DMARC policies.** DMARC enables recipients to authenticate whether an email truly originates from its purported sending domain. However, when a recipient encounters a spoofed or illegitimate email that fails authentication, DMARC relies on the true domain owner to specify a policy for how to treat such email. This design assumes that domain owners will use DMARC policies that result in protective actions, such as QUARANTINE or REJECT. When a domain owner chooses a weaker policy, mail providers deliver the illegitimate email to a user's inbox even if the DMARC authentication fails, in accordance with email standards (RFC 7489 [4]). Unfortunately, prior work has shown that a large number of domains use weak DMARC policies of NONE [50], [51], [55], [67], [68], [69], with roughly two-thirds of the Alexa Top 1M domains employing such a policy (as of May 2020). While poor security hygiene accounts for some of this outcome, many domains choose a weak DMARC enforcement policy for deliverability concerns due to incompatibility with forwarding [55].

Cognizant of this reality, several major email providers have decided to take two types of security actions against email that fails DMARC authentication, regardless of the domain owner's specified policy. First, as noted in prior work [50] and confirmed in our own experiments, Outlook quarantines email if it fails DMARC authentication, even when the email's FROM domain has a weak DMARC policy of NONE. Second, although Gmail, Onet, and Zoho deliver email that fails DMARC authentication to user inboxes, they will display a UI warning to users who read such messages.

These defenses provide protection against attackers who directly send spoofed email to their victims. However, as we will show, email forwarding introduces new complexity that enables attackers to bypass these ad hoc defenses, and thus leverage weak DMARC policies to successfully spoofed email from prominent domains.

**4.2.2. Each domain uses its own infrastructure.** The SPF protocol predates the emergence of large third-party email providers. As a result, SPF implicitly assumes that each organization (domain) maintains its own mailing infrastructure: that the set of authentic server IP addresses specified by a domain's SPF record is not also used by other domains or external users to send email. Unfortu-

8. Our code for automatically sending these messages is available upon request.

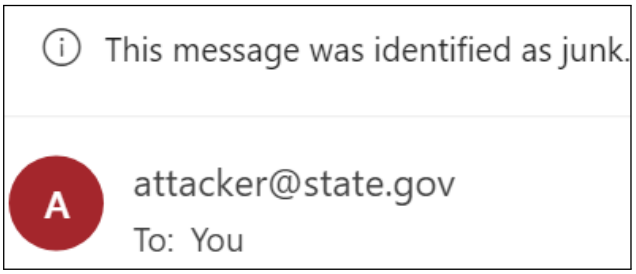

Figure 4: Example message with a FROM header spoofing a domain with DMARC policy REJECT. Outlook delivers it to the spam folder instead of rejecting it.

nately, as documented by Liu et al. [60] and Holzbauer et al. [70], this assumption is invalid today as many organizations outsource their email infrastructure to the *same* third-party providers such as Outlook and Gmail. Hence, all of these domains have delegated the right to send on their behalf to the same third-party — trusting that they will ensure isolation in spite of this blanket authorization.

Concretely, our measurements show that all 16 email providers in our study appear to configure their email infrastructure in this shared fashion. Additionally, at least for email messages forwarded in our experiments, all providers but one (Fastmail) use the same set of servers to send both direct email and forwarded email.

Since SPF no longer provides isolation in this model, the email providers in our study effectively *simulate* it by preventing users from setting arbitrary values in their FROM header. Thus, even though each mail provider is empowered to send any email on behalf of all their mail customers, they prevent customers from taking advantage of this situation by *internally* restricting the FROM headers of outbound email messages coming from a customer's domain. While this defense is effective in the absence of forwarding, we will show how open forwarding mechanisms bypass this filtering (by generating spoofed FROM headers from an *external* server controlled by the adversary), exposing the latent conflict between SPF's design and modern mail service use—ultimately allowing unrestricted email spoofing.

**4.2.3. Quarantining is sufficient.** RFC 7489 [63] suggests that if an email message falls under the scope of a DMARC reject policy, then the receiving server should reject and drop it entirely. However, some providers deviate from this advice by marking it as spam and delivering it to a spam folder, assuming that quarantining a malicious email neutralizes its threat. Our experiments found that five email providers (Outlook, Fastmail, GMX, Inbox.lv and Pobox) adopt this approach. Figure 4 displays an email message from our tests that shows this behavior: it fails DMARC validation, comes from a domain (`state.gov`) that has a DMARC policy of REJECT, but is nonetheless delivered as "spam".

Because these providers quarantine the spoofed email as spam, this design does not appear particularly dangerous.[9] However, as we will show, in combination with email forwarding and another vulnerable feature (per-user domain whitelists in Section 4.2.4), attackers can override this protection and exploit the quarantine-over-reject implementation to spoof email from thousands of popular domains despite their strict DMARC REJECT policy.

**4.2.4. Per-user DMARC overrides are fate-sharing.** Many email providers allow users to override DMARC decisions: users can whitelist domains, and as a result they will still deliver or forward email even if it fails DMARC. Providers offer this flexibility because it can help mitigate errors and improve mail deliverability for the users who need it. However, this feature implicitly assumes that this approach is fate-sharing — that when a user overrides DMARC decisions, the risks of that choice are localized to the individual user account. While true in the single-hop context, forwarding again undermines this assumption. If adversaries can override DMARC decisions on a forwarding account they control, they can use that capability to launder spoofed mail and successfully deliver it downstream.

Based on our measurements, all mail providers support this functionality in some form. Of particular note, four of the five providers mentioned in Section 4.2.3 (Fastmail, GMX, Inbox.lv, and Pobox) allow users to override any DMARC decision for any domains. The fifth (Outlook) allows users to override DMARC decisions for most domains, except for a small set of frequently-spoofed domains that have DMARC policy reject (e.g., `aa.com`) where Outlook appears to apply additional, special protection mechanisms.

For Gmail, Hushmail, iCloud, Mail.ru, Onet, and Zoho, users can override DMARC decisions for domains with DMARC policy NONE or QUARANTINE, but not REJECT. Finally, for Yahoo, we can only override DMARC decisions for domains with a policy of NONE.

## 4.3. Vulnerable Forwarding Features

In the absence of forwarding, the assumptions described above are largely benign and allow the effective blocking of many spoofing attacks. However, when combined with three vulnerable forwarding features, open forwarding, relaxed validation, and unsolicited DKIM signatures, the weaknesses in these assumptions permit several opportunities for bypassing DMARC's protections.

**4.3.1. Open Forwarding.** Many email service providers support a mechanism to automatically forward a user's messages to another account (e.g., to aggregate mail sent to multiple addresses into a single inbox). Because of the prevalence of these common, benign forwarding use cases, many platforms follow a design that we call *open forwarding* (also referred to as "unauthorized forwarding" in previous work [56]). Services with open forwarding allow users to configure their account to forward messages to any destination email address, *without* any verification from the destination address. Open forwarding implicitly assumes users will only forward email to accounts that they control or have a benign relationship with (an assumption that fails when an adversary creates or controls an account entirely for the purpose of malicious forwarding).

9. Some in the mail security industry criticize this weakening of DMARC rules and document attacks that "rescue" such email from the spam folder via social engineering [71], [72].

Our measurements show that open forwarding is still prevalent among providers. Specifically, ten email providers (Outlook, Fastmail, iCloud, Freemail, GoDaddy, Hushmail, Mail2World, Onet, Pobox, and Runbox) allow open forwarding.[10] Moreover, as we demonstrate in three attacks described in Sections 5.1–5.3, when combined with other vulnerabilities, adversaries can exploit open forwarding to attack not only users on those providers that employ this design, but also a broad array of users on other platforms that disallow open forwarding.

**4.3.2. Relaxed Validation.** Since forwarded email can break SPF and DMARC at times, providers may employ relaxed validation for email forwarded by large email providers, assuming that these large providers will prevent spoofed email messages from being forwarded.[11]

We infer that three providers, Gmail, Outlook and Mail.ru, apply some form of relaxed validation. Gmail employs two versions of relaxed validation for forwarded email messages that both (1) fail SPF and DMARC checks and (2) are from domains with a DMARC policy of NONE or QUARANTINE. First, for email messages forwarded via Gmail or Outlook, Gmail delivers them regardless. Second, for messages forwarded via the other providers in our experiments, Gmail delivers the email if it meets specific conditions (more details in Appendix D).

Similarly, our experiments found that Outlook applies relaxed validation for email messages from domains with a DMARC policy of NONE (as discussed in Section 4.2.1, Outlook usually overrides the policy of NONE and quarantines messages that fail DMARC). Specifically, Outlook accepts email messages forwarded via nine major providers (e.g., Gmail and Fastmail), despite failing SPF and DMARC checks. Finally, Mail.ru accepts email messages forwarded via Gmail that fail DMARC from domains with a DMARC policy of NONE or QUARANTINE.

These relaxed validation policies aim to balance the incompatibility of forwarding approaches with anti-spoofing protocols by implicitly trusting high-profile email services. Unfortunately, the complexity introduced by forwarding and its interactions with the diverse set of assumptions we highlight enable attackers to abuse these trust relationships. This is particularly true because all of these providers offer individual consumer accounts. For example, in Section 5.2 we show that an adversary can deliver spoofed email messages from domains that have a DMARC policy of NONE or QUARANTINE to any Gmail user without triggering a warning.

**4.3.3. Unsolicited DKIM Signatures for Hosted Domains.** RFC 6376 [3] and RFC 6377 [73] both recommend that forwarding services apply their own DKIM signatures for forwarded email messages, especially for cases where they modify the message. Shen et al. [56] showed that this configuration can be exploited by a malicious actor via an attack that they called the DKIM Signature Fraud Attack. Specifically, they showed that an adversary can acquire valid DKIM signatures for spoofed email messages if the forwarder naively signs every forwarded email. Such spoofed email messages can successfully pass subsequent DMARC checks if their spoofed sender's domain is the same as the domain used by the forwarding service to sign DKIM signatures. Shen et al. [56] found three providers that had this vulnerable feature: Yahoo, Office365 and Alibaba Cloud.

Through our experiments, we identified that three providers' (iCloud, Hushmail, and Runbox) forwarding implementation contained a variant of this vulnerable feature, which would allow an adversary to mount attacks similar to the DKIM Signature Fraud Attack. Taking iCloud as an example, we find that iCloud adds unsolicited and valid DKIM signatures to spoofed email messages addressed from domains hosted by them. Additionally, iCloud signs the DKIM signature using the same domain as the purported sender's domain in the spoofed email. For instance, iCloud will add a valid DKIM signature signed by the domain `peterborgapps.com` (a domain hosted by iCloud) to spoofed email messages purporting to be from `peterborgapps.com`, allowing the spoofed email messages to pass subsequent DMARC checks. We surmise that providers can add valid DKIM signatures on behalf of hosted domains because they manage DKIM keys for these domains [74], [75].


Figure 5: Example of a successful attack. A spoofed email purporting to be `bush@state.gov` is delivered to a Gmail user's inbox with no warning indicators.

## 5. Attacks

In this section, we demonstrate how an adversary can combine and exploit the issues described in Section 4 to create attacks that reliably bypass existing anti-spoofing protections. In particular, we consider an attack successful if a spoofed email message is delivered to a victim's inbox (i.e., not the spam folder), and yet does not produce a warning to the user. Figure 5 shows an example of a successful attack, where a spoofed email purporting to be from `bush@state.gov` is delivered to a Gmail user's inbox with no warning indication.

We describe four distinct classes of attacks, summarized in Table 3, each of which we have validated empirically using accounts created at the affected providers. Some of these attacks are quite broad — allowing an attacker to spoof email to any email recipient purporting to be from tens of thousands of popular and sensitive domains — while others are more circumscribed in their impact. For each of the attacks described below, we refer to the domain an attacker specifies in their FROM header as the *spoofed domain*. We use the terms *spoofed address* to refer to the full email address appearing in the FROM header and *forwarding domain* to refer to the domain of the forwarder.

10. Mail2World and Pobox do notify the destination account via email about the forwarding setup.

11. Shen et al. [56] also make this observation, but do not document the concrete steps necessary to exploit this vulnerability or demonstrate its practical exploitation.

| | Send email spoofing | Forward via | Deliver to |
|---|---|---|---|
| § 5.1 | Domains with the forwarding domain's SPF information in their SPF records | Six providers including Outlook and iCloud | Any recipient |
| § 5.2 | Arbitrary domains with DMARC policy None or Quarantine | Outlook | Gmail |
| | Arbitrary domains with DMARC policy None | Multiple providers (e.g., Fastmail) | Outlook |
| § 5.3* | Arbitrary domains | Fastmail | Zoho |
| § 5.4 | Domains hosting the mailing list and DMARC policy None | Google Groups, Listserv, Mailman | Any recipient |
| | Arbitrary domains | Gaggle | Any recipient |

* We build on the ARC vulnerability identified by Shen et al. [56], to demonstrate an attack that is practical.

TABLE 3: Summary of email forwarding attacks (§ 5).

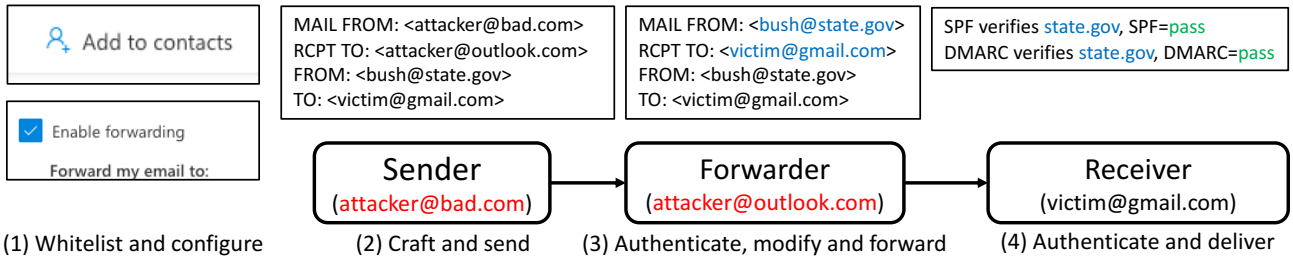


Figure 6: Example of an SPF Incorporation Attack (§ 5.1) exploiting Outlook's open forwarding to spoof email from domains incorporating Outlook's SPF records (e.g., `state.gov`) to arbitrary recipients.

**Threat Models:** For the first three attacks, we assume an adversary controls the sender and forwarding accounts: they possess a server capable of sending spoofed email messages (sender) and a personal account with a specific third-party provider that allows *open forwarding* (forwarder). For the attack described in Section 5.4, we make three assumptions: (a) that adversaries control a malicious server that can send spoofed email messages and try to spoof email from a domain that hosts a mailing list with REM forwarding (e.g., Google Groups, Listserv and Mailman as described in Section 3), (b) that the spoofed domain has a DMARC policy of NONE (all too common); and (c) the sending email address the attacker wishes to impersonate has permission to send to the mailing list.

## 5.1. Exploiting SPF Incorporation

The first attack we describe exploits five discrete issues: three security assumptions (§ 4.2.2, 4.2.3, 4.2.4), the vulnerable *open forwarding* feature that many providers offer (§ 4.3.1), and the header rewriting performed as part of the PMF and MFEF forwarding approaches (§ 3). Crucially, the rise of large third-party email providers violates SPF's assumption that the set of authorized server IP addresses specified by each domain cannot be used by other domains or external users to send email. For example, the owners of domain `state.gov` use Outlook as their email provider. Thus, email messages sent by `state.gov`'s employees will originate from Outlook's mail servers. To ensure reliable delivery, such domains routinely add the server IP addresses of their email provider to their own SPF records. Although intuitive, this configuration creates an overly broad trust assumption: by adding the provider IP addresses to their SPF record, such domains (e.g., `state.gov`) implicitly grant permission for any account hosted by their provider, whether individual or corporate, to send email messages that purportedly come from their domain. This threat is only prevented because large providers like Outlook do not allow users to arbitrarily set or forge their email's FROM header.

However, we observe that by combining header rewriting from PMF and MFEF and the use of *open forwarding*, attackers can overcome this defense and exploit SPF's violated assumption. Specifically, this attack allows an adversary to spoof email from domains that incorporate a third-party provider's SPF information in their own SPF record to any recipient, regardless of the domain's DMARC policy.

**Scope:** This attack works for domains that include the SPF record of any of six large email providers (Outlook, iCloud, Freemail, Hushmail, Mail2World and Runbox) in their own SPF records. Notably, given Outlook's importance as a third-party provider [60], this attack allows an attacker to spoof email on behalf of tens of thousands of popular domains.

Indeed, over 12% of the Alexa 100K most popular domains are vulnerable as a result (and almost 8% of the top 1M domains). A cursory examination of this list identified a range of potentially sensitive domains such as those hosting large news reporting organizations (e.g., `washingtonpost.com`, `latimes.com`, and `apnews.com`), financial services (e.g., `mastercard.com`, `transunion.com`, and `docusign.com`), domain registrars (e.g., `godaddy.com`), certificate authorities (e.g., `sectigo.com` and `digicert.com`) and large law firms (e.g., `perkinscoie.com`). In addition, 32% of US `.gov` domains are vulnerable (including 22% of the domains used by Federal agencies). At the Federal level this includes the majority of US cabinet organizations (e.g., `state.gov`, `dhs.gov` and `doe.gov`), a range of security sensitive agencies (e.g., `odni.gov`, `cisa.gov` and `secretservice.gov`) as well as those charged with public health and safety (such as `fema.gov`, `nih.gov`, and `cdc.gov`). At the state and local level, virtually all primary state government domains (e.g., `mass.gov`) are

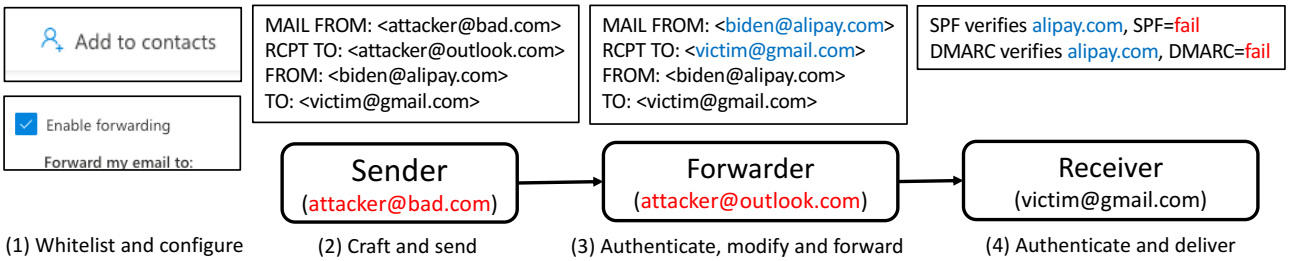


Figure 7: Example of a spoofed email attack exploiting open forwarding and relaxed validation for forwarded email from well-known providers (§ 5.2). Note that the spoofed domain, `alipay.com`, has a DMARC policy of Quarantine and thus should not be delivered.

vulnerable (including a broad range of congress, judiciary, and law enforcement domains in each state) and over 40% of all `.gov` domains used by cities.[12]

**Example:** Figure 6 shows an example of this attack using Outlook as the forwarding service. An attacker starts by creating a personal account for forwarding (`attacker@outlook.com`), adding the spoofed address (`bush@state.gov`) to the account's "allowlist" (thereby preventing any quarantining by Outlook), and configuring the account to forward all email to the desired target (`victim@gmail.com`). In this case, the spoofed domain `state.gov` includes Outlook's SPF record (`spf.protection.outlook.com`) into its own SPF record and has a DMARC policy of REJECT. Next, the attacker forges an email that purportedly originates from `state.gov` and sends it to their personal Outlook account. Normally, Outlook would quarantine this email because it fails DMARC validation (§ 4.2.3). However, since the spoofed address is present in the account's allowlist, this configuration overwrites the quarantine decision (§ 4.2.4), and as a result, Outlook would forward the spoofed email to the target.

As per Outlook's MFEF forwarding implementation, MAIL FROM is rewritten to match the FROM header, `bush@state.gov` in our example. Finally, the recipient's mail server receives the forwarded email and performs authentication checks. From the recipient's perspective, the spoofed email passes SPF validation because the MAIL FROM domain (`state.gov`) lists Outlook's SPF information in its SPF record, and the forwarding configuration arranged by the attacker ensures that the recipient receives this spoofed email from Outlook's servers. Moreover, this attack also ensures that DMARC's alignment check succeeds because the MAIL FROM and FROM domain are both `state.gov`. We validated this attack in practice, consistently sending spoofed email messages such as the example shown in Figure 5 to our own Gmail account, where it was delivered to the inbox without warning.[13] In addition to Outlook, this attack also succeeds with iCloud, Freemail, Hushmail, Mail2World and Runbox.

## 5.2. Abusing Relaxed Forwarding Validation

The second attack exploits the fact that many email providers apply relaxed validation policies to forwarded mail (§ 4.3.2), particularly when messages arrive from well-known mail providers. When combined with open forwarding, an attacker can abuse this behavior to spoof email from any domain that has a DMARC policy of QUARANTINE (or NONE) to any mail server that applies these relaxed measures (e.g., Gmail and Outlook). Recall that, in the absence of forwarding, attackers cannot spoof email from a domain with a DMARC policy of QUARANTINE. Provider-specific defenses, such as when Outlook quarantines any email that fails DMARC (§ 4.2.1), will also stop such direct, single-hop attacks.

**Scope:** As described earlier in Section 4.3.2, Gmail and Outlook use relaxed validation checks for forwarded email. We find that an adversary can mount this attack against users with Gmail/Outlook email accounts as well as users who use GSuite and Outlook 365 for email services.[14]

**Example:** Figure 7 illustrates the steps of this attack using an example where the adversary creates a personal Outlook account to forward spoofed email messages to Gmail recipients. First, the adversary selects a spoofed email address from a domain with a DMARC policy of QUARANTINE or NONE (we use `alipay.com` in this example, a prominent Chinese payment company), adds the address to their forwarding account's allowlist, and configures their forwarding account to send email to the victim (recipient). Like the first attack, the attacker then sends a message from this spoofed address to their forwarding account, which is then forwarded to the recipient.

When the final recipient's mail server receives the email, the server will observe that the email comes from a "well-known" provider, apply its relaxed validation checks, and successfully deliver the email to the recipient's inbox (even though the spoofed email fails normal SPF and DMARC checks).[15]

12. We have not broadly examined domains representing government offices outside the US, but we note that both `gchq.gov.uk` and `ncsc.gov.uk` are also vulnerable.

13. Note that we did discover some exceptions in our experiments. For a small set of high-profile domains that have a DMARC policy of Reject (e.g., `aa.com`, `foxnews.com` and `ikea.com`), Outlook would quarantine spoofed email regardless of whether users have added the spoofed address to their account's allowlist (Section 4.2.3). We surmise that Outlook applies special protections for a set of high-profile or frequently spoofed domains.

14. Mail.ru also uses relaxed validation, but since it is only applied to email forwarded via Gmail, which does not allow open forwarding, this attack does not work for Mail.ru.

15. Additionally, we note that Gmail would usually display a UI warning for forwarded email messages. However, no UI warning is displayed for this email due to a bug detailed in Appendix B.2.

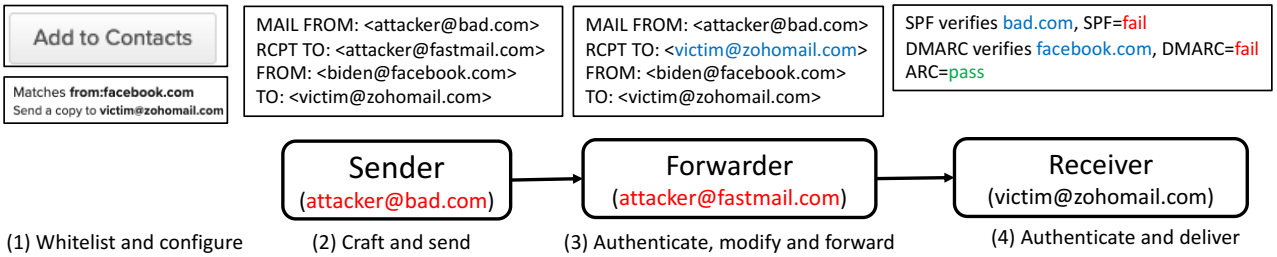


Figure 8: Example attack that exploits Zoho's vulnerable ARC implementation and open forwarding to spoof email from arbitrary domains to any Zoho recipient (§ 5.3).

### 5.3. Targeting ARC Vulnerabilities

The third attack allows an adversary to deliver spoofed email messages from arbitrary domains to Zoho users. This attack exploits Zoho's vulnerable implementation of the experimental Authenticated Received Chain (ARC) protocol [76], which was first documented by Shen et al. [56]. Due to this bug, Zoho incorrectly reads ARC headers and will deliver arbitrary email messages with ARC headers added by providers such as Gmail and Fastmail to the recipient's inbox without any warning. However, we show that this issue is not limited to interactions between Gmail and Zoho customers. We demonstrate how further issues, including the fact that Zoho trusts and (incorrectly) reads ARC headers added by Fastmail (Appendix A), open forwarding (§ 4.3.1), and several forwarding assumptions (§ 4.2.3, § 4.2.4), can be combined with the underlying ARC vulnerability to allow an adversary to deliver spoofed email messages from arbitrary domains to arbitrary Zoho users.

This attack again highlights the fact that email security protocols are distributed and independently-configured components, where vulnerable decisions by one party incur harm to downstream recipients but not necessarily to their own users. Notably, the actions taken by one provider (e.g., Fastmail) can unexpectedly undermine the security of users on another platform (e.g., Zoho).

**Scope:** Our experiments show that this attack can target arbitrary users of Zoho, which is estimated to have more than 10 million users [77].

**Example:** Figure 8 shows the mechanics of this attack in the context of an attacker with a forwarding account on Fastmail who targets a recipient on Zoho.

First, the adversary creates a Fastmail account for forwarding, adds their spoofed address (`biden@facebook.com`) to their allowlist, and configures their account to forward all mail to the target user at Zoho (`victim@zohomail.com`). Second, the adversary crafts and sends spoofed email from their own servers (e.g., `attacker@bad.com`) to their forwarding account at Fastmail. Third, although this email will fail anti-spoofing validation, Fastmail will still faithfully forward it to the target user at Zoho due to the sender's presence on the user's allowlist (exploiting the security assumption discussed in § 4.2.4). As part of the forwarding process, Fastmail will modify the RCPT TO header and add corresponding ARC headers to the spoofed email. Finally, upon receiving the forwarded email, Zoho's mail server will perform DMARC validation. Although the spoofed email will fail SPF and DMARC checks, Zoho's vulnerable ARC implementation will misinterpret the ARC headers that Fastmail attached (Appendix A). As a result, Zoho will treat the email as passing DMARC and deliver the spoofed message to the victim's inbox. We end by noting that this attack would not have worked for domains with DMARC policy REJECT had Fastmail rejected spoofed email messages addressed from such domains (§ 4.2.3).

### 5.4. Abusing Mailing Lists

The final attack allows an adversary to abuse the forwarding process used by mailing lists so that spoofed email, which would otherwise fail DMARC authentication checks, successfully passes both SPF and DMARC validation. This attack targets domains with a weak DMARC policy of NONE, and exploits the way in which many mailing lists rewrite email headers during their forwarding process. Concretely, this attack allows an adversary to abuse REM header rewriting (§ 3) to launder spoofed email through mailing lists such that the forwarded email appears as if it originated from the legitimate sender, even though the original email fails DMARC authentication.[16]

**Scope:** Our experiments show that attackers can conduct this attack across all four popular mailing list services: Google Groups, Mailman, Listserv, and Gaggle.

This attack only affects organizations that use a mailing list configured under their own domain name, and with a DMARC policy of NONE for their (sub)domain. While these requirements appear restrictive, prior work has found that many organizations (such as major U.S. universities) have exactly this configuration [55]. Indeed, in querying `.edu` and `.gov` domains, roughly 10% of all `.edu` domains and 5% of all `.gov` domains are potentially susceptible to this attack.[17]

Additionally, for mailing lists like Gaggle that do not enforce DMARC checks before forwarding (Appendix B.1), this attack affects every organization using their services, even when the domain has adopted stronger DMARC policies.

16. One such attack used to distribute phishing messages at our institution was part of the impetus for this study.

17. As examples, Yale University operates `yale.edu` with a DMARC policy of NONE and hosts multiple mailing lists using Mailman, and the State of Washington operates a range of mailing lists using Listserv and whose `wa.gov` domain also has a DMARC policy of NONE.

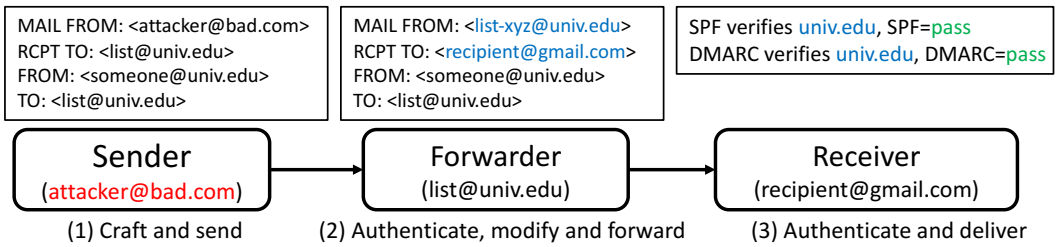


Figure 9: Spoofed email attack that abuses mailing lists like Google Groups (§ 5.4).

**Example:** Figure 9 describes an example of this attack using Google Groups. First, an attacker selects a target email address (`someone@univ.edu`) to impersonate in their spoofed email, and sends the spoofed message from their malicious server to the organization's mailing list (`list@univ.edu`). Although the email fails DMARC validation, the mailing list will still accept the message (because `univ.edu` has a DMARC policy of NONE). As part of REM forwarding, the mailing list will rewrite the MAIL FROM header such that its domain matches the mailing list's domain, and then forward the email to the list's members. As a result, when a recipient's server receives the message it will successfully pass SPF validation since the domain of the rewritten MAIL FROM (`univ.edu`) allows the mailing list to send on its behalf. Moreover, the spoofed message will also pass DMARC alignment checks, since the rewriting performed during REM forwarding ensures that the MAIL FROM and FROM domains are identical.[18]

During our experiments, we also observed that some mailing list services, such as Gaggle, do not enforce DMARC policies at all (Appendix B.1). This lack of enforcement allows the attack to succeed regardless of the spoofed domain's DMARC policy. We provide more details in Appendix F.

## 6. Ethics and Disclosure

When sending spoofed email messages in our experiments, we took deliberate steps to avoid impacting any real users. First, we only sent spoofed email messages to accounts that we created ourselves. Second, we initially tested each attack by spoofing domains that we created and controlled for this research. Once we established that our attacks could succeed using these test domains, we ran a small set of experiments that spoofed email from real domains (to validate the absence of any unforeseen protection); however, these email messages were only sent to our test accounts and did not spoof existing, legitimate email addresses from these domains. Finally, all of our email messages contained innocuous text (e.g., "a spoofed email") that would not themselves cause harm.

We have disclosed all of the vulnerabilities and attacks to the affected providers. As of the time of publication, we have received affirmative feedback from all affected providers and we summarize our current understanding of their present state here. Zoho has not only patched the issue with their ARC implementation (also confirmed by Wang et al. [57], who conducted their measurements after the patch) and awarded us a bug bounty, but is also further augmenting the security of its ARC implementation. Microsoft confirmed the vulnerabilities (with severity "Important", the highest severity assigned to email spoofing bugs) and awarded us a bug bounty. They have partially fixed the issues by rejecting spoofed email messages purporting to be from domains that have a DMARC policy of REJECT [78]. Gaggle confirmed the issues we flagged and stated that they would start enforcing DMARC. Gmail fixed the issues we reported. iCloud partially fixed the issues we reported by not forwarding email messages that fail DMARC authentication (except for domains with DMARC policy NONE). Hushmail fixed the issues we reported by not forwarding email messages that fail DMARC authentication. Freemail fixed the issues we reported by not forwarding spoofed email messages from domains that are their customers. Mail2World attempted to fix the issues by using spam filters and remains vulnerable. Runbox did not view the issues we reported as vulnerabilities. Instead, they consider monitoring account activities post-complaints sufficient.

18. Note that while our examples show this attack using the organization's top-level domain, it is also effective for any of the organization's subdomains (if the subdomains also have DMARC policy NONE) due to DMARC's inherent relaxed alignment policy.

## 7. Discussion and Mitigation

We end by summarizing the root causes of the issues we discovered, and discuss potential mitigation strategies.

### 7.1. Discussion

In this work, we examine the complexities introduced by email forwarding to email security. We identify a diverse set of email forwarding mechanisms, assumptions, and features, and demonstrate how they can be combined together to perform evasion attacks. These attacks highlight four fundamental issues.

First, as already demonstrated in prior work and further highlighted in our paper, email security involves distributed, optional, and independently-configured components implemented by different parties. In such an architecture, the "authenticity" of an email is commonly determined by the party with the weakest security settings. While traditionally email is sent directly from sender to receiver, forwarding involves three parties instead of two and introduces an extra layer of complexity. As we have shown, a vulnerable forwarder can jeopardize the security of downstream recipients that do not have problematic configurations or implementations. This inversion of incentives and capabilities naturally complicates mitigating forwarding vulnerabilities.

A second problem is that email forwarding has never been fully standardized, despite the longevity and popularity of its use. A lack of standardization has led to ad-hoc implementation decisions, each making different assumptions. This ad-hoc nature of implementations makes it challenging to perform both manual security analysis (analyzing individual implementation decisions is a non-trivial task even for experts) and automated testing (any such tool needs to account for the specific implementations of each provider). While our large-scale empirical measurements have been able to reveal the assumptions made by providers and their implications, it has required substantial manual work. This manual process is a reflection of the fact that there exists no unified framework or standard for implementing email forwarding.

A third issue is that email is a large, slowly-evolving ecosystem with a wide range of legacy systems and protocols that need to be accommodated. One example we highlight is the "outdated" assumption made by SPF (§ 4.2.2). When SPF was first designed in the early 2000s, it was common practice for each domain owner to maintain their own mail infrastructure. However, this assumption is obsolete in the modern era, as many domains outsource their email services to third-party providers such as Outlook and Google [60]. These large providers often share the same email infrastructure across all customers (both business and personal accounts), violating the assumptions made by SPF. To mitigate the risks this reality poses to SPF, providers usually prevent users from setting arbitrary values in their FROM header. However, past literature has shown that this defense is not always implemented correctly [5]. We build on top of this prior work by identifying a new attack that can circumvent existing defenses through forwarding (§ 5.1).

Last but not least, the intrinsic nature of email forwarding is to transparently send an existing message to a new address "on behalf" of its original recipient — a goal very much at odds with the anti-spoofing function of protocols such as SPF and DMARC. As such, a range of ad-hoc decisions have been made to increase the deliverability of forwarded email messages, such as using the REM+MOD forwarding mechanism (§ 3), treating forwarded messages specially (§ 4.3.2), and adding DKIM signatures to forwarded messages (§ 4.3.3). As we have demonstrated, these decisions can fail to foresee unexpected interactions that lead to vulnerabilities, even with a lot of deliberation.

## 7.2. Mitigation

The attacks we demonstrate highlight the complicated interactions between email forwarding and existing anti-spoofing mechanisms. We start by reviewing short-term mitigations that could reduce some of the most significant risks we have uncovered. We then discuss challenges in developing more comprehensive solutions, which would require significant changes in either protocol or operational practices.

A core issue we highlight in this paper is the ability to forward spoofed email messages to arbitrary recipients, a critical element in each of the first three attacks in Section 5. To mitigate this issue, providers could either block spoofed email messages from being forwarded, or enforce that a forwarder can only forward to accounts under their control by requiring explicit confirmation (similar to the online domain validation used by modern certificate authorities). However, we note that either approach comes with a usability tradeoff, and different providers make choices based on their considerations. Indeed, providers like Gmail and Mail.ru opted for the former option, while others like iCloud and Hushmail opted for the latter.

As well, we advocate that providers should enforce a domain's DMARC REJECT policy when specified, rather than substituting a weaker policy. If Outlook rejected spoofed email messages from such domains, the impact of the first attack exploiting SPF incorporation would narrow substantially. We understand that Outlook has plans to take such action in the future [78].

Unfortunately, all the defenses described above reflect a case of misaligned incentives: the recipients of spoofed email (e.g., spam and phishing) cannot implement this change, but instead need to rely on the entire ecosystem of providers and forwarding services to adopt such defenses.

Email providers can also mitigate the second attack (§ 5.2) by eliminating relaxed validation policies. This approach would protect their users from receiving spoofed email without relying on changes by other platforms or services. However, to prevent benign forwarding from breaking will likely require providers to then implement ARC validation (which in turn places ARC implementation requirements on external forwarders).

For the final attack (§ 5.4) that exploits mailing lists, potential mitigations trade usability for security. First, list owners can turn on message moderation and set their mailing lists to be private. While these measures increase the difficulty of performing email spoofing attacks, they do not rule out the attack entirely. A dedicated attacker might nonetheless identify a member of the mailing list and craft an email that fools a list's moderator. Second, some mailing list services, such as Listserv, support confirm-before-send [79], which requests confirmation from the (true) sender address before delivery. While this mechanism would impose significant overheads in general, these costs might be acceptable by limiting this confirmation requirement to incoming email that fails DMARC authentication checks.

In addition to the short-term mitigations mentioned above that are specific to forwarding, others [5], [56] have proposed solutions such as improving UI notification, building better testing tools, and revising RFC standards, which are also important to consider. Additionally, the newly proposed ARC protocol may also help mitigate some of the issues we have uncovered. However, ARC is still in the early stages of development and deployment, its details are yet to be fleshed out and its effectiveness in practice remains to be seen.

Lastly, we note that comprehensively fixing email forwarding would require a more fundamental set of changes (e.g., redesigning the entire suite of email security protocols), which will face significant deployment challenges given the current state of the email ecosystem. Chief among these challenges is that any new solution designed to fix forwarding must address backwards compatibility, a task complicated by email's forty-year-old ecosystem of varied protocols, implementations and use cases. Specifically, one must carefully consider how

any new approach interacts and interoperates with existing systems (e.g., mail providers and filtering service providers) and protocols (e.g., SPF, DKIM and DMARC). While security might be enhanced by embracing a single standard approach to forwarding (e.g., when a message should be forwarded, what forwarding mechanisms should be used, what information should be added to forwarded messages, and how the receiving account should be verified), any such choice will inevitably align well with certain providers and conflict with those whose existing services have made different choices or who operate under different threat models. Finally, it is not enough to merely standardize new protocols, but one must then also incentivize and coordinate their universal deployment and operation. Thus, while such an aspirational goal is worthy of attention, it seems likely that email will continue to benefit from incremental and reactive improvements, such as those discussed earlier, for some time yet.

## 8. Conclusion

Internet-based email has been in use since the early 1970s and the SMTP protocol has been in use since 1980. It is arguably the longest-lived text-based communication system in wide use. Unsurprisingly, its design did not anticipate the range of challenges we face today and, because of its central role, we have been forced to upgrade email protocols slowly and with deference to a wide range of legacy systems and expectations. Perhaps nowhere is this more clear than around the issue of authentication. Email protocols have no widely-used mechanism for establishing the authenticity of sender addresses, and thus we have focused on authenticating the domain portion of the email address (largely motivated by spam and phishing).

In this work, using large-scale empirical measurements of 20 prominent email forwarding services, we identify a diverse set of email forwarding mechanisms, assumptions, and features, and demonstrate how they can be combined together to perform four types of evasion attacks. While we are the first academic paper to document these attacks, retrospectively examining Mailop [80], a prominent mailing list for mail operators, we have also found traces [81] of real-world attacks that are similar to what we reported in this paper.

The attacks we document exploit four kinds of problems. One fundamental issue is that email security protocols are distributed, optional, and independently-configured components. This creates a large and complex attack surface with many possible interactions that cannot be easily anticipated or administered by any single party. A second problem is that email forwarding was never standardized, leading to ad-hoc implementation decisions that might be vulnerable. A third problem is that protocol assumptions for SPF are grounded at a point in time and have not been updated as practices have changed. Domains now out-source their mail service to large providers that share mail infrastructure across customers, undermining assumptions made in the design of SPF. Lastly, the intrinsic nature of email forwarding is to transparently send an existing message to a new address "on behalf" of its original recipient. This creates complex chain-of-trust issues that are at odds with implicit assumptions that mail is sent directly from sender to receiver. Indeed, it is this complication that has driven the creation of ARC.

While there are certain short-term mitigations (e.g., eliminating the use of open forwarding) that will significantly reduce the exposure to the attacks we have described here, ultimately email requires a more solid security footing if it is to effectively resist spoofing attacks going forwards.

## Acknowledgments

We thank our anonymous reviewers for their insightful and constructive suggestions and feedback. We thank Nishant Bhaskar, Cindy Moore, and Jennifer Folkestad for their operational support. We thank Stewart Grant for proofreading the paper. We thank Brad Chen for collecting feedback from Google. We thank John Levine and Weihaw Chuang for their comments on the paper. Funding for this work was provided in part by National Science Foundation grants CNS-1705050 and CNS-2152644, the UCSD CSE Postdoctoral Fellows program, the Irwin Mark and Joan Klein Jacobs Chair in Information and Computer Science, and operational support from the UCSD Center for Networked Systems as well as the Twente University Centre for Cybersecurity Research (TUCCR).

## References

[1] Verizon, "2021 Data Breach Investigations Report," 06 2021. [Online]. Available: https://enterprise.verizon.com/resources/reports/2021-data-breach-investigations-report.pdf

[2] S. Kitterman, "Sender Policy Framework (SPF) for Authorizing Use of Domains in Email," RFC 7208, 04 2014. [Online]. Available: https://rfc-editor.org/rfc/rfc7208.txt

[3] M. Kucherawy, D. Crocker, and T. Hansen, "DomainKeys Identified Mail (DKIM) Signatures," RFC 6376, 09 2011. [Online]. Available: https://rfc-editor.org/rfc/rfc6376.txt

[4] M. Kucherawy and E. Zwicky, "Domain-based Message Authentication, Reporting, and Conformance (DMARC)," RFC 7489, 03 2015. [Online]. Available: https://rfc-editor.org/rfc/rfc7489.txt

[5] J. Chen, V. Paxson, and J. Jiang, "Composition Kills: A Case Study of Email Sender Authentication," in *Proceedings of the 2020 USENIX Security Symposium*, 2020, pp. 2183–2199.

[6] J. C. Klensin, "Simple Mail Transfer Protocol," RFC 5321, 08 2008. [Online]. Available: https://rfc-editor.org/rfc/rfc5321.txt

[7] P. Resnick, "Internet Message Format," RFC 5322, 08 2008. [Online]. Available: https://www.rfc-editor.org/info/rfc5322

[8] PowerDMARC. (2023, 04) What is a DMARC Policy? [Online]. Available: https://powerdmarc.com/what-is-dmarc-policy/

[9] EasyDMARC. (2023, 04) What is a DMARC Policy? [Online]. Available: https://easydmarc.com/blog/what-is-a-dmarc-policy/

[10] Wikipedia. (2021, 05) Electronic Mailing List. [Online]. Available: https://en.wikipedia.org/wiki/Electronic_mailing_list

[11] Proofpoint. (2021, 05) Secure Email Gateways Definition and Comparison. [Online]. Available: https://www.proofpoint.com/us/threat-reference/email-gateway

[12] A. Kishore. (2021, 05) The Best Way to Switch to a New Email Address. [Online]. Available: https://www.online-tech-tips.com/computer-tips/switch-to-a-new-email-address/

[13] S. Abu-Nimeh, D. Nappa, X. Wang, and S. Nair, "A Comparison of Machine Learning Techniques for Phishing Detection," in *Proceedings of the 2007 Anti-phishing Working Groups 2nd Annual Ecrime Researchers Summit*, 2007, pp. 60–69.

[14] A. Bergholz, J. H. Chang, G. Paass, F. Reichartz, and S. Strobel, "Improved Phishing Detection using Model-Based Features," in *Proceedings of the 2008 Conference on Email and Anti-Spam*, 2008.

[15] I. Fette, N. Sadeh, and A. Tomasic, "Learning to Detect Phishing Emails," in *Proceedings of the 2007 International Conference on World Wide Web*, 2007, pp. 649–656.

[16] S. Garera, N. Provos, M. Chew, and A. D. Rubin, "A Framework for Detection and Measurement of Phishing Attacks," in *Proceedings of the 2007 ACM Workshop on Recurring Malcode*, 2007, pp. 1–8.

[17] C. Whittaker, B. Ryner, and M. Nazif, "Large-Scale Automatic Classification of Phishing Pages," in *Proceedings of the 2010 Network and Distributed System Security Symposium*, 2010.

[18] S. Duman, K. Kalkan-Cakmakci, M. Egele, W. Robertson, and E. Kirda, "Emailprofiler: Spearphishing Filtering with Header and Stylometric Features of Emails," in *Proceedings of the 2016 IEEE Annual Computer Software and Applications Conference*, 2016, pp. 408–416.

[19] M. Khonji, Y. Iraqi, and A. Jones, "Mitigation of Spear Phishing Attacks: A Content-based Authorship Identification Framework," in *Proceedings of the 2011 International Conference for Internet Technology and Secured Transactions*, 2011, pp. 416–421.

[20] M. Zhao, B. An, and C. Kiekintveld, "Optimizing Personalized Email Filtering Thresholds to Mitigate Sequential Spear Phishing Attacks," in *Proceedings of the 2016 AAAI Conference on Artificial Intelligence*, 2016.

[21] G. Stringhini and O. Thonnard, "That Ain't You: Blocking Spearphishing Through Behavioral Modelling," in *Proceedings of the 2015 International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment*, 2015, pp. 78–97.

[22] G. Ho, A. Cidon, L. Gavish, M. Schweighauser, V. Paxson, S. Savage, G. M. Voelker, and D. Wagner, "Detecting and Characterizing Lateral Phishing at Scale," in *Proceedings of the 2019 Security Symposium*, 2019, pp. 1273–1290.

[23] G. Ho, A. Sharma, M. Javed, V. Paxson, and D. Wagner, "Detecting Credential Spearphishing in Enterprise Settings," in *Proceedings of the 2017 USENIX Security Symposium*, 2017, pp. 469–485.

[24] A. Cidon, L. Gavish, I. Bleier, N. Korshun, M. Schweighauser, and A. Tsitkin, "High Precision Detection of Business Email Compromise," in *Proceedings of the 2019 USENIX Security Symposium*, 2019, pp. 1291–1307.

[25] X. Han, N. Kheir, and D. Balzarotti, "Phisheye: Live Monitoring of Sandboxed Phishing Kits," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, 2016, pp. 1402–1413.

[26] J. Onaolapo, E. Mariconti, and G. Stringhini, "What Happens after You Are Pwnd: Understanding the Use of Leaked Webmail Credentials in the Wild," in *Proceedings of the 2016 Internet Measurement Conference*, 2016, pp. 65–79.

[27] K. Thomas, F. Li, C. Grier, and V. Paxson, "Consequences of Connectivity: Characterizing Account Hijacking on Twitter," in *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, 2014, pp. 489–500.

[28] E. Bursztein, B. Benko, D. Margolis, T. Pietraszek, A. Archer, A. Aquino, A. Pitsillidis, and S. Savage, "Handcrafted Fraud and Extortion: Manual Account Hijacking in the Wild," in *Proceedings of the 2014 Conference on Internet Measurement Conference*, 2014, pp. 347–358.

[29] E. Lastdrager, I. C. Gallardo, P. Hartel, and M. Junger, "How Effective is Anti-Phishing Training for Children?" in *Proceedings of the 2017 Symposium on Usable Privacy and Security*, 2017, pp. 229–239.

[30] B. Reinheimer, L. Aldag, P. Mayer, M. Mossano, R. Duezguen, B. Lofthouse, T. Von Landesberger, and M. Volkamer, "An Investigation of Phishing Awareness and Education over Time: When and How to Best Remind Users," in *Proceedings of the 2020 Symposium on Usable Privacy and Security*, 2020, pp. 259–284.

[31] P. Mayer, D. Poddebniak, K. Fischer, M. Brinkmann, J. Somorovsky, A. Sasse, S. Schinzel, and M. Volkamer, ""I don't know why I check this..."-Investigating Expert Users' Strategies to Detect Email Signature Spoofing Attacks," in *Proceedings of the 2022 Symposium on Usable Privacy and Security*, 2022, pp. 77–96.

[32] D. D. Caputo, S. L. Pfleeger, J. D. Freeman, and M. E. Johnson, "Going Spear Phishing: Exploring Embedded Training and Awareness," *IEEE Security and Privacy*, vol. 12, no. 1, pp. 28–38, 2013.

[33] E. Spero and R. Biddle, "Out of Sight, Out of Mind: UI Design and the Inhibition of Mental Models of Security," in *Proceedings of the 2020 New Security Paradigms Workshop*, 2020, pp. 127–143.

[34] S. Sheng, M. Holbrook, P. Kumaraguru, L. F. Cranor, and J. Downs, "Who Falls for Phish? A Demographic Analysis of Phishing Susceptibility and Effectiveness of Interventions," in *Proceedings of the 2010 SIGCHI Conference on Human Factors in Computing Systems*, 2010, pp. 373–382.

[35] P. Kumaraguru, S. Sheng, A. Acquisti, L. F. Cranor, and J. Hong, "Teaching Johnny Not to Fall for Phish," *ACM Transactions on Internet Technology*, vol. 10, no. 2, 2010.

[36] J. Müller, M. Brinkmann, D. Poddebniak, H. Böck, S. Schinzel, J. Somorovsky, and J. Schwenk, ""Johnny, you are fired!"– Spoofing OpenPGP and S/MIME Signatures in Emails," in *Proceedings of the 2019 USENIX Security Symposium*, 2019, pp. 1011–1028.

[37] D. Poddebniak, C. Dresen, J. Müller, F. Ising, S. Schinzel, S. Friedberger, J. Somorovsky, and J. Schwenk, "Efail: Breaking S/MIME and OpenPGP Email Encryption using Exfiltration Channels," in *Proceedings of the 2018 USENIX Security Symposium*, 2018, pp. 549–566.

[38] J. Schwenk, M. Brinkmann, D. Poddebniak, J. Müller, J. Somorovsky, and S. Schinzel, "Mitigation of Attacks on Email End-to-end Encryption," in *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, 2020, pp. 1647–1664.

[39] J. Müller, M. Brinkmann, D. Poddebniak, S. Schinzel, and J. Schwenk, "Mailto: Me Your Secrets. On Bugs and Features in Email End-to-end Encryption," in *Proceedings of the 2020 IEEE Conference on Communications and Network Security*, 2020, pp. 1–9.

[40] C. Stransky, O. Wiese, V. Roth, Y. Acar, and S. Fahl, "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*, 2022.

[41] H. Lee, M. I. Ashiq, M. Müller, R. van Rijswijk-Deij, T. Chung *et al.*, "Under the Hood of DANE Mismanagement in SMTP," in *Proceedings of the 2022 USENIX Security Symposium*, 2022, pp. 1–16.

[42] H. Lee, A. Gireesh, R. van Rijswijk-Deij, T. Chung *et al.*, "A Longitudinal and Comprehensive Study of the DANE Ecosystem in Email," in *Proceedings of the 2020 USENIX Security Symposium*, 2020.

[43] Z. Durumeric, D. Adrian, A. Mirian, J. Kasten, E. Bursztein, N. Lidzborski, K. Thomas, V. Eranti, M. Bailey, and J. A. Halderman, "Neither Snow Nor Rain Nor MITM...: An Empirical Analysis of Email Delivery Security," in *Proceedings of the 2015 Internet Measurement Conference*, 2015, pp. 27–39.

[44] I. D. Foster, J. Larson, M. Masich, A. C. Snoeren, S. Savage, and K. Levchenko, "Security by Any Other Name: On the Effectiveness of Provider Based Email Security," in *Proceedings of the 2015 ACM SIGSAC Conference on Computer and Communications Security*, 2015, pp. 450–464.

[45] D. Poddebniak, F. Ising, H. Böck, and S. Schinzel, "Why TLS is Better Without STARTTLS: A Security Analysis of STARTTLS in the Email Context," in *Proceedings of the 2021 USENIX Security Symposium*, 2021, pp. 4365–4382.

[46] R. Holz, J. Amann, O. Mehani, M. Wachs, and M. A. Kaafar, "TLS in the Wild: An Internet-wide Analysis of TLS-based Protocols for Electronic Communication," *arXiv preprint arXiv:1511.00341*, 2015.

[47] W. Mayer, A. Zauner, M. Schmiedecker, and M. Huber, "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*. IEEE, 2016, pp. 10–20.

[48] S. Ullrich. (2021, 06) Breaking DKIM — on Purpose and by Chance. [Online]. Available: https://noxxi.de/research/breaking-dkim-on-purpose-and-by-chance.html

[49] S. Haddouche. (2021, 06) Mailsploit. [Online]. Available: https://mailsploit.pwnsdx.com/index

[50] H. Hu and G. Wang, "End-to-End Measurements of Email Spoofing Attacks," in *Proceedings of the 2018 USENIX Security Symposium*, 2018, pp. 1095–1112.

[51] D. Tatang, F. Zettl, and T. Holz, "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*, 2021, pp. 354–369.

[52] C. Deccio, T. Yadav, N. Bennett, A. Hilton, M. Howe, T. Norton, J. Rohde, E. Tan, and B. Taylor, "Measuring Email Sender Validation in the Wild," in *Proceedings of the 2021 International Conference on Emerging Networking Experiments and Technologies*, 2021, pp. 230–242.

[53] C. Wang, K. Shen, M. Guo, Y. Zhao, M. Zhang, J. Chen, B. Liu, X. Zheng, H. Duan, Y. Lin *et al.*, "A Large-scale and Longitudinal Measurement Study of DKIM Deployment," in *Proceedings of the 2022 USENIX Security Symposium*, 2022, pp. 1185–1201.

[54] N. Bennett, R. Sowards, and C. Deccio, "Spfail: Discovering, Measuring, and Remediating Vulnerabilities in Email Sender Validation," in *Proceedings of the 22nd ACM Internet Measurement Conference*, 2022, pp. 633–646.

[55] H. Hu, P. Peng, and G. Wang, "Towards Understanding the Adoption of Anti-Spoofing Protocols in Email Systems," in *Proceedings of the 2018 IEEE Cybersecurity Development*, 2018, pp. 94–101.

[56] K. Shen, C. Wang, M. Guo, X. Zheng, C. Lu, B. Liu, Y. Zhao, S. Hao, H. Duan, Q. Pan, and M. Yang, "Weak Links in Authentication Chains: A Large-scale Analysis of Email Sender Spoofing Attacks," in *Proceedings of the 2021 USENIX Security Symposium*, 2021, pp. 3201–3217.

[57] C. Wang and G. Wang, "Revisiting Email Forwarding Security under the Authenticated Received Chain Protocol," in *Proceedings of the 2022 ACM Web Conference*, 2022, pp. 681–689.

[58] K. Andersen, B. Long, S. Blank, and M. Kucherawy, "The Authenticated Received Chain (ARC) Protocol," RFC 8617, 07 2019. [Online]. Available: https://rfc-editor.org/rfc/rfc8617.txt

[59] EasyDMARC. (2022, 05) Email forwarding and dmarc dkim spf. [Online]. Available: https://easydmarc.com/blog/email-forwarding-and-dmarc-dkim-spf/

[60] E. Liu, G. Akiwate, M. Jonker, A. Mirian, S. Savage, and G. M. Voelker, "Who's Got Your Mail? Characterizing Mail Service Provider Usage," in *Proceedings of the 2021 ACM Internet Measurement Conference*, 2021, pp. 122–136.

[61] Wikipedia. (2021, 03) Email forwarding. [Online]. Available: https://en.wikipedia.org/wiki/Email_forwarding

[62] M. W. Wong. (2003, 06) Sender Rewriting Scheme. [Online]. Available: http://www.open-spf.org/svn/project/specs/drafts/draft-mengwong-sender-rewrite-01.txt

[63] B. Leiba and W. Segmuller, "Sieve Email Filtering: Relational Extension," RFC 5231, Jan. 2008. [Online]. Available: https://rfc-editor.org/rfc/rfc5231.txt

[64] J. R. Levine and R. Gellens, "Mailing Lists and Non-ASCII Addresses," RFC 6783, 11 2012. [Online]. Available: https://rfc-editor.org/rfc/rfc6783.txt

[65] SpamResource. (2021, 06) Google groups rewriting from addresses to handle dmarc policy. [Online]. Available: https://www.spamresource.com/2014/04/google-groups-rewriting-from-addresses.html

[66] Postfix. (2021, 06) The Postfix Home Page. [Online]. Available: http://www.postfix.org/

[67] A. Portier, H. Carter, and C. Lever, "Security in Plain TXT," in *Proceedings of the 2019 International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment*, 2019, pp. 374–395.

[68] S. Maroofi, M. Korczynski, and A. Duda, "From Defensive Registration to Subdomain Protection: Evaluation of Email Anti-Spoofing Schemes for High-Profile Domains," in *Proceedings of the 2020 Network Traffic Measurement and Analysis Conference*, 2020.

[69] S. Maroofi, M. Korczyński, A. Hölzel, and A. Duda, "Adoption of Email Anti-Spoofing Schemes: A Large Scale Analysis," *IEEE Transactions on Network and Service Management*, vol. 18, no. 3, pp. 3184–3196, 3 2021.

[70] F. Holzbauer, J. Ullrich, M. Lindorfer, and T. Fiebig, "Not That Simple: Email Delivery in the 21st Century," in *Proceedings of the 2022 USENIX Annual Technical Conference*, 2022, pp. 295–308.

[71] L. Ovadia. (2021, 04) Microsoft O365 Fails to Block Spoofed Emails Sent from Microsoft.com. [Online]. Available: https://ironscales.com/blog/Microsoft-O365-Fails-to-Block-Spoofed-Emails/

[72] E. Montalbano. (2021, 04) Spearphishing Attack Spoofs Microsoft.com to Target 200M Office 365 Users. [Online]. Available: https://threatpost.com/spearphishing-attack-spoofs-microsoft-office-365/162001/

[73] M. Kucherawy, "DomainKeys Identified Mail (DKIM) and Mailing Lists," RFC 6377, Sep. 2011. [Online]. Available: https://www.rfc-editor.org/info/rfc6377

[74] Apple. (2022, 10) Set up an existing domain with icloud mail. [Online]. Available: https://support.apple.com/en-us/HT212524

[75] Hushmail. (2023, 04) Setting dkim and dmarc for your domain. [Online]. Available: https://help.hushmail.com/hc/en-us/articles/360043601032-Setting-DKIM-and-DMARC-for-your-domain

[76] A. Specification. (2021, 03) ARC Specification for Email. [Online]. Available: http://arc-spec.org/

[77] P. Murugan. (2021, 06) Celebrating 10 years of Zoho Mail with 10 Million+ business users. [Online]. Available: https://www.zoho.com/blog/mail/celebrating-10-years-of-zoho-mail.html

[78] J. Dellapina. (2023, 04) [mailop] hotmail will start rejecting messages that fail dmarc. [Online]. Available: https://www.mail-archive.com/mailop@mailop.org/msg18539.html

[79] Indiana University. (2021, 06) On my LISTSERV list, what does the Send keyword do, and what are the possible settings? [Online]. Available: https://kb.iu.edu/d/alml

[80] Mailop. (2022, 10) Mailop. [Online]. Available: https://www.mailop.org/

[81] ——. (2022, 10) Re: [mailop] spoofed message passing dmarc. [Online]. Available: https://www.mail-archive.com/mailop@mailop.org/msg16129.html

[82] Fastmail. (2021, 04) Sender Authentication. [Online]. Available: https://www.fastmail.help/hc/en-us/articles/1500000280461-Sender-authentication#arc

| Received by | Added by: Gmail | Zoho | Fastmail | Pobox |
|---|---|---|---|---|
| Gmail | ✓ | | ✓ | ✓ |
| Outlook | ✓ | | | |
| Zoho | ✓ | | ✓ | ✓ |
| Fastmail | ✓ | ✓ | ✓ | ✓ |
| Pobox | ✓ | | ✓ | ✓ |

TABLE 4: Trust of ARC headers between providers.

## A. ARC Adoption in the Wild

Five email providers (Gmail, Outlook, Zoho, Fastmail, and Pobox) and two mailing list services (Google Groups and Mailman) implement ARC validation. Because ARC is still an experimental protocol, many email providers only evaluate ARC headers added by a small set of other providers whom they trust [82]. Based on measurements through test accounts that we created, Table 4 shows the ARC trust relationships among the providers we tested: Gmail trusts ARC headers added by Fastmail, Pobox and itself; Outlook trusts ARC headers added by Gmail; Zoho trusts ARC headers added by Gmail, Fastmail, Pobox and itself; Fastmail trusts ARC headers added by Gmail, Zoho, Pobox and itself; and Pobox trusts ARC headers added by Gmail, Fastmail and itself.

We also note two details. First, we cannot test whether other providers trust ARC headers added by Outlook. ARC headers are only evaluated when a forwarded email fails DMARC authentication checks. However, in our experiments, Outlook only adds ARC headers to certain email messages that will pass DMARC authentication checks after forwarding. This design prevents us from testing which providers trust Outlook's ARC headers. Second, our results suggest that Zoho does not add ARC headers when forwarding messages internally between Zoho accounts, so we leave that cell empty in the table.

## B. Additional Implementation Errors

We detail two other implementation errors identified during our measurement. These two errors play minor roles in the attacks we discover.

### B.1. Ignoring Security Protocols

Systems that assume every email is legitimate tend not to enforce DMARC. Instead, they will permissively accept email messages, even if they fail DMARC authentication checks and the purported FROM domains have a stricter DMARC policy of QUARANTINE or REJECT. Such assumptions can create serious security issues.

From our experiments, we found one mailing list provider (Gaggle) and three mail providers (Freemail, Mail2World and Runbox) that do not enforce DMARC. Beyond delivering spoofed email to users and allowing spoofed email to be forwarded, we show later (Section 5.4) that incorrect enforcement, combined with the standard forwarding modifications that Gaggle applies to email headers, allows an attacker to abuse Gaggle. Spoofed email messages that initially fail DMARC authentication will receive a fresh set of headers that correctly pass SPF and DMARC validation after they are forwarded through a Gaggle-operated mailing list.


Figure 10: Gmail annotating the sending address of an email.

### B.2. Gmail UI Bug

After a receiver accepts and processes email, the user's mail user agent (MUA) displays the message for viewing. Thus, MUAs and their UI warnings serve as the last line of defense against spoofed email messages. However, previous work [50], [56], [5] has found multiple security issues in various MUAs, especially on mobile platforms. In our experiments, we focus on the native MUAs (web interfaces) provided by the nine email platforms in our study. These MUAs not only have widespread usage, as the default MUA for many users, but are also maintained by the email providers and tend to have better security practices.

Among all native MUAs, only Gmail, Onet and Zoho have implemented warning systems that display UI indicators when an email is forwarded or fails DMARC authentication. Gmail, for instance, annotates the sending address (e.g., `adminrec@univ.edu` *via* `e2ma.net` as shown in Figure 10).

However, we observed a bug in Gmail's warning system for a subset of forwarded messages. In particular, Gmail does not display an indicator for a forwarded email message if (1) it does not contain any DKIM headers, and (2) it has the same domain in both the MAIL FROM and FROM headers. This policy does not pose a problem in single-sender email settings, because adversaries still need to bypass SPF and DMARC. However, we present a new attack that uses this bug in conjunction with forwarding and other vulnerabilities to deliver spoofed email messages that look no different than legitimate messages (Section 5.2).

## C. Additional Attack Screenshots

We ran a small set of experiments that spoofed email impersonating real domains to validate the attacks described in Section 5. Our experiments confirmed that these attacks succeed. Below, we present the screenshots of spoofed email messages successfully delivered to users' inboxes.

For the attack described in Section 5.2, Figure 11 shows a spoofed message forwarded via a personal Outlook account to a Gmail account we created, and delivered to the recipient's inbox without any security warnings. The spoofed address in this example impersonates a sender at `alipay.com` (a prominent Chinese payment company with a DMARC policy of QUARANTINE).

For the attack described in Section 5.3, Figure 12 illustrates that this attack succeeds without any security

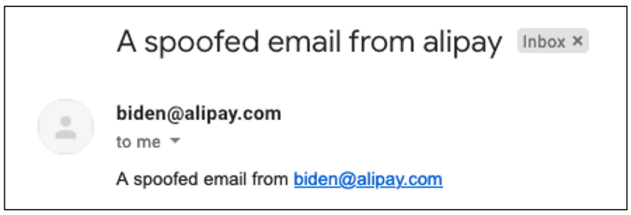


Figure 11: Email spoofing `biden@alipay.com` via Outlook.


Figure 12: Email spoofing `biden@facebook.com` via Fastmail.


Figure 13: Spoofed email message taking advantage of Outlook's relaxed forwarding validation policy.

warnings, even though the spoofed domain in our experiment, `facebook.com`, has a DMARC policy of REJECT.

## D. Additional Details for the Attack in Section 5.2

This section makes three additional observations about the attacks on providers with relaxed forwarding validation as described in Section 5.2.

First, in addition to forwarding from personal Outlook accounts, an adversary can also forward from personal accounts with other providers (e.g., Fastmail) to Gmail recipients. As mentioned earlier in Section 4.3.2, there is an additional caveat: the TO header of the spoofed email cannot be the same as victim's email address. An astute recipient might see that the TO field corresponds to someone else's email account, and become suspicious about the email's validity. To reduce suspicion in this case, the adversary can set the human-readable name portion of the email's TO header to "me" or the victim's name (while keeping the address different).

Second, adversaries need to leverage popular mail providers as forwarders in this attack; they cannot exploit relaxed forwarding validation by using their own servers as forwarders. In our experiments, we tested using both a personal Outlook account as well as a mail server we controlled as forwarders. The first version results in successful attacks delivered to user inboxes, but the latter did not. We suspect this outcome is because our mail server domain has lower reputation than Outlook's mail servers.

Finally, in Section 5.2, we demonstrate the attack against a recipient using Gmail. A similar attack can be mounted against Outlook recipients by forwarding via a personal Fastmail account. This attack allows an adversary to spoof email messages from many domains that have a DMARC policy NONE to arbitrary Outlook recipients.[19] Figure 13 shows a spoofed email message forwarded via a personal account to an Outlook account we created, and delivered without any security warnings. The spoofed FROM header in this example impersonates a sender at `lesechos.fr` (a French financial newspaper with a DMARC policy of NONE).

19. Similar to the caveats described in Section 4.2.3, we observe that Outlook applies additional restrictions to a small set of high-profile domains with a DMARC policy of NONE (e.g., `citizensbank.com`), which blocks the delivery of spoofed emails from these domains.

## E. Additional Details for the Attack in Section 5.3

Adversaries can broaden the scope of the attack described in Section 5.3 by using a forwarding account at any email provider that Zoho trusts for ARC purposes, including Gmail, and routing their spoofed email through multiple forwarding hops. In particular, an attacker can obtain ARC headers in one forwarding hop via Gmail, and then bypass Gmail's lack of open forwarding by forwarding the email through a second account that does allow open forwarding (e.g., Outlook). For example, first, the attacker would send their spoofed email message to their Gmail account, which they configured to forward to their malicious Outlook account. During the forwarding process Gmail will attach a set of ARC headers to the email message. Next, the spoofed email will arrive at their malicious Outlook account, which then forwards the email to any arbitrary Zoho recipient (because Outlook supports open forwarding). This forwarded email message will contain Gmail's attached ARC headers, enabling the attack to successfully pass DMARC validation checks as a result of Zoho's vulnerable ARC implementation.

Using our test accounts, we validated that this multi-hop attack variation successfully delivers spoofed messages to the inbox of a Zoho recipient without any warnings.

## F. Additional Details for the Attack in Section 5.4

In addition to the attacks described in Section 5.4, we found additional attack variants related to Gaggle. Figure 14 shows an example of an attack that abuses Gaggle's use of REM + MOD forwarding (Section 3). This attack works regardless of the DMARC policy of the spoofed address's domain. First, an attacker chooses

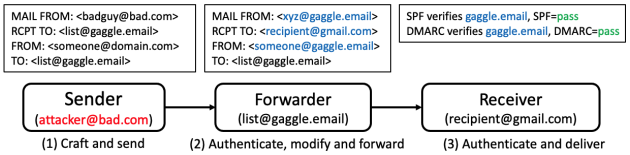


Figure 14: Attack flow for Gaggle.

an address to spoof (`someone@foo.com`) that is allowed to send to a mailing list on a vulnerable provider (`list@gaggle.email`), and sends a spoofed email message purporting to come from that address. This spoofed email will fail DMARC validation, but because Gaggle does not enforce DMARC (Appendix B.1), it will forward the email to the mailing list's recipients as normal (Stage 2). Since Gaggle uses a REM + MOD forwarding process, it will rewrite the MAIL FROM header to use the mailing list's domain (e.g., a new MAIL FROM address of `xyz@gaggle.email`). Finally, when the spoofed email message arrives at the recipient's mail server, it will properly pass SPF validation and DMARC alignment checks: the rewritten MAIL FROM domain allows the mailing list to send on its behalf, and the domain matches the FROM address's domain (`gaggle.email`).

Additionally, we note that mailing list software such as Listserv and Mailman require a backend MTA. In our experiments we used Postfix with DMARC turned on, a configuration which follows good security practice. However, in practice many organizations might not use this configuration because many MTAs (including Postfix) do not enforce DMARC by default. In these cases, the attacker can spoof email from any target domain, regardless of its DMARC policy, much like the attack against Gaggle.