End-to-End Measurements of Email Spoofing Attacks

Hang Hu and Gang Wang, ***Virginia Tech***

https://www.usenix.org/conference/usenixsecurity18/presentation/hu

This paper is included in the Proceedings of the 27th USENIX Security Symposium.

August 15–17, 2018 • Baltimore, MD, USA

ISBN 978-1-939133-04-5

Open access to the Proceedings of the 27th USENIX Security Symposium is sponsored by USENIX.

End-to-End Measurements of Email Spoofing Attacks

Hang Hu
Virginia Tech
hanghu@vt.edu

Gang Wang
Virginia Tech
gangwang@vt.edu

Abstract

Spear phishing has been a persistent threat to users and organizations, and yet email providers still face key challenges to authenticate incoming emails. As a result, attackers can apply spoofing techniques to impersonate a trusted entity to conduct highly deceptive phishing attacks. In this work, we study *email spoofing* to answer three key questions: (1) How do email providers detect and handle forged emails? (2) Under what conditions can forged emails penetrate the defense to reach user inbox? (3) Once the forged email gets in, how email providers warn users? Is the warning truly effective?

We answer these questions by conducting an end-to-end measurement on 35 popular email providers and examining user reactions to spoofing through a real-world spoofing/phishing test. Our key findings are three folds. *First*, we observe that most email providers have the necessary protocols to detect spoofing, but still allow forged emails to reach the user inbox (*e.g.*, Yahoo Mail, iCloud, Gmail). *Second*, once a forged email gets in, most email providers have no warning for users, particularly for mobile email apps. Some providers (*e.g.*, Gmail Inbox) even have misleading UIs that make the forged email look authentic. *Third*, a few email providers (9/35) have implemented visual security indicators on unverified emails. Our phishing experiment shows that security indicators have a positive impact on reducing risky user actions, but cannot eliminate the risk. Our study reveals a major miscommunication between email providers and end-users. Improvements at both ends (server-side protocols and UIs) are needed to bridge the gap.

1 Introduction

Despite the recent development of the system and network security, human factors still remain a weak link. As a result, attackers increasingly rely on phishing tactics to breach various target networks [62]. For example, email phishing has involved in nearly half of the 2000+ reported security breaches in recent two years, causing a leakage of billions of user records [4].

Email spoofing is a critical step in phishing, where the attacker impersonates a trusted entity to gain the victim's trust. According to the recent report from the Anti-Phishing Working Group (APWG), email spoofing is widely in spear phishing attacks to target employees of various businesses [2]. Unfortunately, today's email transmission protocol (SMTP) has no built-in mechanism to prevent spoofing [56]. It relies on email providers to implement SMTP extensions such as SPF [40], DKIM [19] and DMARC [50] to authenticate the sender. Since implementing these extensions is *voluntary*, their adoption rate is far from satisfying. Real-world measurements conducted in 2015 have shown that among Alexa top 1 million domains, 40% have SPF, 1% have DMARC, and even fewer are correctly/strictly configured [23, 27].

The limited server-side protection is likely to put users in a vulnerable position. Since not every sender domain has adopted SPF/DKIM/DMARC, email providers still face key challenges to reliably authenticate all the incoming emails. When an email failed the authentication, it is a "blackbox" process in terms of how email providers handle this email. Would forged emails still be delivered to users? If so, how could users know the email is questionable? Take Gmail for example, Gmail delivers certain forged emails to the inbox and places a security indicator on the sender icon (a red question mark, Figure 6(a)). We are curious about how a broader range of email providers handle forged emails, and how much the security indicators actually help to protect users.

In this paper, we describe our efforts and experience in evaluating the real-world defenses against email spoofing[1]. We answer the above questions through empirical end-to-end spoofing measurements, and a user study.

[1] Our study has been approved by our local IRB (IRB-17-397).

First, we conduct measurements on how popular email providers detect and handle forged emails. The key idea is to treat each email provider as a blackbox and vary the input (forged emails) to monitor the output (the receiver's inbox). Our goal is to understand under what conditions the forged/phishing emails are able to reach the user inbox and what security indicators (if any) are used to warn users. *Second*, to examine how users react to spoofing emails and the impact of security indicators, we conduct a real-world phishing test in a user study. We have carefully applied "deception" to examine users' natural reactions to the spoofing emails.

Measurements. We start by scanning Alexa top 1 million hosts from February 2017 to January 2018. We confirm that the overall adoption rates of SMTP security extensions are still low (SPF 44.9%, DMARC 5.1%). This motivates us to examine how email providers handle incoming emails that failed the authentication.

We conduct end-to-end spoofing experiments on 35 popular email providers used by billions of users. We find that forged emails can penetrate the majority of email providers (34/35) including Gmail, Yahoo Mail and Apple iCloud under proper conditions. Even if the receiver performs all the authentication checks (SPF, DKIM, DMARC), spoofing an unprotected domain or a domain with "relaxed" DMARC policies can help the forged email to reach the inbox. In addition, spoofing an "existing contact" of the victim also helps the attacker to penetrate email providers (*e.g.*, Hotmail).

More surprisingly, while most providers allow forged emails to get in, rarely do they warn users of the unverified sender. Only 9 of 35 providers have implemented some security indicators: 8 providers have security indicators on their *web interface* (*e.g.*, Gmail) and only 4 providers (*e.g.*, Naver) have the security indicators consistently for the *mobile apps*. There is no security warning if a user uses a third-party email client such as Microsoft Outlook. Even worse, certain email providers have misleading UI elements which help the attacker to make forged emails look authentic. For example, when attackers spoof an existing contact (or a user from the same provider), 25 out of 35 providers will automatically load the spoofed sender's photo, a name card or the email history along with the forged email. These UI designs are supposed to improve the email usability, but in turn, help the attacker to carry out the deception when the sender address is actually spoofed.

Phishing Experiment. While a handful of email providers have implemented security indicators, the real question is how effective they are. We answer this question using a user study ($N = 488$) where participants examine spoofed phishing emails with or without security indicators on the interface. This is a real-world phishing test where deception is carefully applied such that users examine the spoofed emails without knowing that the email is part of an experiment (with IRB approval). We debrief the users and obtain their consent after the experiment.

Our result shows that security indicators have a positive impact on reducing risky user actions but cannot eliminate the risk. When a security indicator is not presented (the controlled group), out of all the users that opened the spoofed email, 48.9% of them eventually clicked on the phishing URL in the email. For the other group of users to whom we present the security indicator, the corresponding click-through rate is slightly lower (37.2%). The impact is consistently positive for users of different demographics (age, gender, education level). On the other hand, given the 37.2% click-through rate, we argue that the security indicator cannot eliminate the phishing risk. The server-side security protocols and the user-end security indicators should be both improved to maximize the impact.

Contributions. We have 3 key contributions:

- *First*, our end-to-end measurement provides new insights into how email providers handle forged emails. We reveal the trade-offs between email availability and security made by different email providers
- *Second*, we are the first to empirically analyze the usage of security indicators on spoofed emails. We show that most email providers not only lack the necessary security indicators (particularly on mobile apps), but also have misleading UIs that help the attackers.
- *Third*, we conduct a real-world phishing test to evaluate the effectiveness of the security indicator. We demonstrate the positive impact (and potential problems) of the security indicator and provide the initial guidelines for improvement.

The quantitative result in this paper provides an end-to-end view on how spoofed emails could penetrate major email providers and all the way affect the end users. We hope the results can draw more attention from the community to promoting the adoption of SMTP security extensions. In addition, we also seek to raise the attention of email providers to designing and deploying more effective UI security indicators, particularly for the less protected mobile email apps. We have communicated the results with the Gmail team and offered suggestions to improve the security indicators.

2 Background and Methodology

Today's email system is built upon the SMTP protocol, which was initially designed without security in mind.

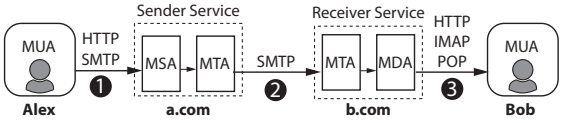


Figure 1: Email transmission from Alex to Bob.

Security extensions were introduced later to provide confidentiality, integrity, and authenticity. Below, we briefly introduce SMTP and related security extensions. Then we introduce our research questions and methodology.

2.1 SMTP and Email Spoofing

Simple Mail Transfer Protocol (SMTP) is an Internet standard for electronic mail transmission [56]. Figure 1 shows the three main steps to deliver an email message. (❶) Starting from the sender's Mail User Agent (MUA), the message is first transmitted to the Mail Submission Agent (MSA) of the sender's service provider via STMP or HTTP/HTTPS. (❷) Then the sender's Mail Transfer Agent (MTA) sends the message to the receiver's email provider using SMTP. (❸) The message is then delivered to the receiving user by the Mail Delivery Agent (MDA) via Internet Message Access Protocol (IMAP), Post Office Protocol (POP) or HTTP/HTTPS.

When initially designed, SMTP did not have any security mechanisms to authenticate the sender identity. As a result, attackers can easily craft a forged email to impersonate/spoof an arbitrary sender address by modifying the "`MAIL FROM`" field in SMTP. Email spoofing is a critical step in a phishing attack — by impersonating a trusted entity as the email sender, the attacker has a higher chance to gain the victim's trust. In practice, attackers usually exploit SMTP in step (❷) by setting up their own MTA servers.

Alternatively, an attacker may also exploit step (❶) if a legitimate email service is not carefully configured. For example, if `a.com` is configured as an open relay, attacker can use `a.com`'s server and IP to send forged emails that impersonate *any* email addresses.

2.2 Email Authentication

To defend against email spoofing attacks, various security extensions have been proposed and standardized including SPF, DKIM and DMARC. There are new protocols such as BIMI and ARC that are built on top of SPF, DKIM, and DMARC. In this paper, we primarily focus on SPF, DKIM, and DMARC since they have some level of adoption by email services in practice. BIMI and ARC have not been fully standardized yet, and we will discuss them later in §7.

SPF. Sender Policy Framework (SPF) allows an email service (or an organization) to publish a list of IPs that are authorized to send emails for its domain (RFC7208 [40]). For example, if a domain "`a.com`" published its SPF record in the DNS, then the receiving email services can check this record to match the sender IP with the sender email address. In this way, only authorized IPs can send emails as "`a.com`". In addition, SPF allows the organization to specify a policy regarding how the receiver should handle the email that failed the authentication.

DKIM. DomainKeys Identified Mail (DKIM) uses the public-key based approach to authenticate the email sender (RFC6376 [19]). The sender's email service will place a digital signature in the email header signed by the private key associated to the sender's domain. The receiving service can retrieve the sender's public key from DNS to verify the signature. In order to query a DKIM public key from DNS, one not only needs the domain name but also a *selector* (an attribute in the DKIM signature). Selectors are used to permit multiple keys under the same domain for more a fine-grained signatory control. DKIM does not specify what actions that the receiver should take if the authentication fails.

DMARC. Domain-based Message Authentication, Reporting and Conformance (DMARC) is built on top of SPF and DKIM (RFC7489 [50]), and it is not a standalone protocol. DMARC allows the domain administrative owner to publish a policy to specify what actions the receiver should take when the incoming email fails the SPF and DKIM check. In addition, DMARC enables more systematic reporting from receivers to senders. A domain's DMARC record is available under `_dmarc.domain.com` in DNS.

2.3 Research Questions and Method

Despite the available security mechanisms, significant challenges remain when these mechanisms are not properly deployed in practice. Measurements conducted in 2015 show that the adoption rates of SMTP security extensions are far from satisfying [23, 27]. Among Alexa top 1 million domains, only 40% have published an SPF record, and only 1% have a DMARC policy. These results indicate a real challenge to protect users from email spoofing. First, with a large number of domains not publishing an SPF/DKIM record, email providers cannot reliably detect incoming emails that spoof unprotected domains. Second, even a domain is SPF/DKIM-protected, the lack of (strict) DMARC policies puts the receiving server in a difficult position. It is not clear how the email providers at the receiving end would handle unverified emails. Existing works [23, 27] mainly focus on the authentication protocols on the *server-side*. However, there is still a big gap between the server-side detection and the actual impact on users.

Status	All Domain # (%)	MX Domain # (%)
Total domains	1,000,000 (100%)	792,556 (100%)
w/ SPF	492,300 (49.2%)	473,457 (59.7%)
w/ valid SPF	**448,741 (44.9%)**	**430,504 (54.3%)**
Policy: soft fail	272,642 (27.3%)	268,317 (33.9%)
Policy: hard fail	**125,245 (12.5%)**	**112,415 (14.2%)**
Policy: neutral	49,798 (5.0%)	48,736 (6.1%)
Policy: pass	1,056 (0.1%)	1,036 (0.1%)
w/ DMARC	51,222 (5.1%)	47,737 (6.0%)
w/ valid DMARC	**50,619 (5.1%)**	**47,159 (6.0%)**
Policy: none	39,559 (4.0%)	36,984 (4.7%)
Policy: reject	**6,016 (0.6%)**	**5,225 (0.7%)**
Policy: quarantine	5,044 (0.5%)	4,950 (0.6%)

Table 1: SPF/DMARC statistics of Alexa 1 million domains. The data was collected in January 2018.

Our Questions. Our study seeks to revisit the email spoofing problem by answering three key questions. (1) When email providers face uncertainty in authenticating incoming emails, how would they handle the situation? Under what conditions would forged emails be delivered to the users? (2) Once forged emails reach the inbox, what types of warning mechanisms (if any) are used to notify users of the unverified sender address? (3) How effective is the warning mechanism? Answering these questions is critical to understanding the *actual risks* exposed to users by spoofing attacks.

We answer question(1)–(2) through end-to-end spoofing experiments (§3, §4 and §5). For a given email provider, we treat it as a "blackbox". By controlling the input (*e.g.*, forged emails) and monitoring the output (receiver's inbox), we infer the decision-making process inside the blackbox. We answer question(3) by conducting a large user study (§6). The idea is to let users read spoofing/phishing emails with and without security indicators.

Ethics. We have taken active steps to ensure research ethics. Our measurement study only uses dedicated email accounts owned by the authors and there is no real user getting involved. In addition, to minimize the impact on the target email services, we have carefully controlled the message sending rate (one message every 10 minutes), which is no different than a regular email user. For the user study that involves "deception", we worked closely with IRB for the experiment design. More detailed ethical discussions are presented later.

3 Adoption of SMTP Extensions

The high-level goal of our measurement is to provide an end-to-end view of email spoofing attacks against popular email providers. Before doing so, we first examine the recent adoption rate of SMTP security extensions compared with that of three years ago [23, 27]. This helps to provide the context for the challenges that email providers face to authenticate incoming emails.

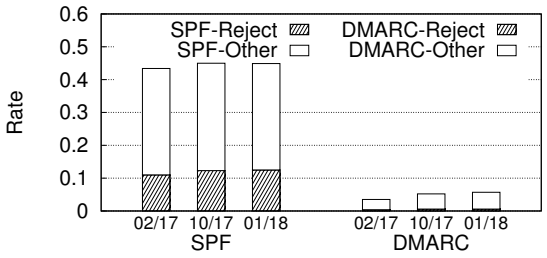


Figure 2: The adoption rate of SPF and DMARC among Alexa 1 million domains across three snapshots.

Scanning Alexa Top 1 Million Domains. Email authentication requires the sender domains to publish their SPF/DKIM/DMARC records to DNS. To examine the recent adoption rate of SPF and DMARC, we crawled 3 snapshots the DNS record for Alexa top 1 million hosts [1] in February 2017, October 2017, and January 2018. Similar to [23, 27], this measurement cannot apply to DKIM, because querying the DKIM record requires knowing the *selector* information for every each domain. The selector information is only available in the DKIM signature in the email header, which is not a public information. We will measure the DKIM usage later in the end-to-end measurement.

Recent Adoption Rates. Table 1 shows the statistics for the most recent January 2018 snapshot. SPF and DMARC both have some increase in the adoption rate but not very significant. About 44.9% of the domains have published a valid SPF record in 2018 (40% in 2015 [27]), and 5.1% have a valid DMARC record in 2018 (1.1% in 2015 [27]). The invalid records are often caused by the domain administrators using the wrong format for the SPF/DMARC record. Another common error is to have multiple records for SPF (or DMARC), which is equivalent to "no record" according to RFC7489 [50]. Figure 2 shows the adoption rate for all three snapshots. Again, the adoption rates have been increasing at a slow speed.

Among the 1 million domains, 792,556 domains are MX domains (*i.e.*, mail exchanger domains that host email services). The adoption rates among MX domains are slightly higher (SPF 54.3%, DMARC 6.0%). For non-MX domains, we argue that it is also important to publish the SPF/DMARC record. For example, `office.com` is not a MX domain, but it hosts the website of Microsoft Office. Attackers can spoof `office.com` to phish Microsoft Office users or even the employees.

Failing Policy. SPF and DMARC both specify a policy regarding what actions the receiver should take after the authentication fails. Table 1 shows that only a small portion of the domains specifies a strict "reject" policy: 12.5% of the domains set "hard fail" for SPF, and

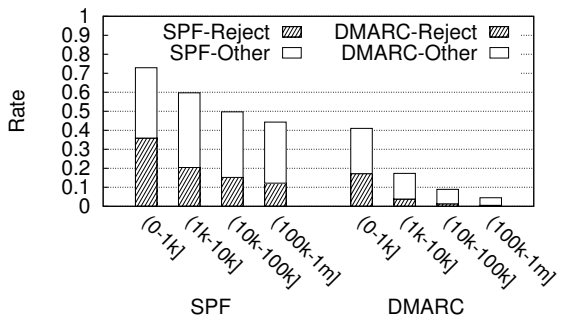


Figure 3: The adoption rate as a function of the domains' Alexa rankings (January 2018).

0.6% set "reject" for DMARC. The rest of the domains simply leave the decision to the email receiver. "Soft fail"/"quarantine" means that the email receiver should process the email with caution. "Neutral"/"none" means that no policy is specified. SPF's "pass" means that the receiver should let the email go through. If a domain has both SPF and DMARC policies, DMARC overwrites SPF as long as the DMARC policy is not "none".

Domains that use DKIM also need to publish their policies through DMARC. The fact that only 5.1% of the domains have a valid DMARC record and 0.6% have a "reject" policy indicates that most DKIM adopters also did not specify a strict reject policy.

Popular Domains. Not too surprisingly, popular domains' adoption rates are higher as shown in Figure 3. We divide the top 1 million domains into log-scale sized bins. For SPF, the top 1,000 domains have an adoption rate of 73%. For DMARC, the adoption rate of top 1000 domains is 41%. This indicates that administrators of popular domains are more motivated to prevent their domains from being spoofed. Nevertheless, there is still a large number of (popular) domains remain unprotected.

4 End-to-End Spoofing Experiments

Given the current adoption rate of SMTP extension protocols, it is still challenging for email providers to reliably authenticate all incoming emails. When encountering questionable emails, we are curious about how email providers make such decisions. In the following, we describe the details of our measurement methodology and procedures.

4.1 Experiment Setup

We conduct end-to-end spoofing experiments on popular email providers that are used by billions of users. As shown in Figure 4, for a given email provider (`B.com`), we set up a user account under `B.com` as the email receiver (`test@B.com`). Then we set up an experimental

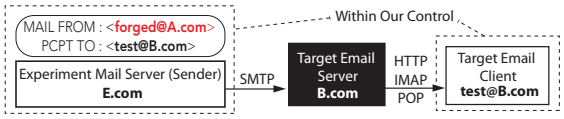


Figure 4: End-to-end spoofing experiment setup. We use our server `E.com` to send a forged email to the target email service `B.com` by spoofing `A.com`.

server (`E.com`) to send forged emails to the receiver account. Our server runs a Postfix mail service [3] to directly interact with the target mail server using SMTP. By controlling the input (the forged email) and observing the output (the receiver account), we infer the decision-making process inside of the target email service.

Selecting Target Email Providers. This study focuses on popular and public email services with two considerations. First, popular email services such as Yahoo Mail and Gmail are used by more than one billion users [46, 55]. Their security policies and design choices are likely to impact more people. Second, to perform end-to-end experiments, we need to collect data from the receiver end. Public email services allow us to create an account as the receiver. Our experiment methodology is applicable to private email services but requires collaborations from the internal users.

To obtain a list of popular public email services, we refer to Adobe's leaked user database (152 million email addresses, 9.3 million unique email domains) [41]. We ranked the email domains based on popularity, and manually examined the top 200 domains (counting for 77.7% of all email addresses). After merging domains from the same service (*e.g.*, `hotmail.com` and `outlook.com`) and excluding services that don't allow us to create an account, we obtained a short list of 28 email domains. To include the more recent public email services, we searched on Google and added 6 more services (`yeah.net`, `protonmail.com`, `tutanota.com`, `zoho.com`, `fastmail.com`, and `runbox.com`). We notice that Google's Gmail and Inbox have very different email interfaces and we treat them as two services.

In total, we have 35 popular email services which cover 99.8 million email addresses (65.7%) in the Adobe database. As an additional reference, we also analyze the Myspace database (131.4 million email addresses) [54]. We find that 101.8 million email addresses (77.5%) are from the 35 email services, confirming their popularity. The list of the email providers is shown in Table 2

4.2 Experiment Parameters

To examine how different factors affect the outcome of email spoofing, we apply different configurations to the experiment. We primarily focus on parameters that

are likely to affect the spoofing outcome, including the spoofed sender address, email content, sender IP, and the receiver's email client (user interface).

Spoofed Sender Address. The sender address is a critical part of the authentication. For example, if the spoofed domain (`A.com`) has a valid SPF/DKIM/DMARC record, then the receiver (in theory) is able to detect spoofing. We configure three profiles for the spoofed sender domain: (1) *None*: no SPF/DKIM/DMARC record (*e.g.*, `thepiratebay.org`); (2) *Relaxed*: SPF/DKIM with a "none" policy (*e.g.*, `tumblr.com`); and (3) *Strict*: SPF/DKIM with a strict "reject" policy (*e.g.*, `facebook.com`). For each profile, we randomly pick 10 domains (30 domains in total) from Alexa top 5000 domains (the detailed list is in Appendix A).

Email Content. Email content can affect how spam filters handle the incoming emails [11]. Note that our experiment is not to reverse-engineer exactly how spam filters weight different keywords, which is an almost infinite searching space. Instead, we focus on spoofing (where the sender address is forged). We want to *minimize* the impact of spam filters and examine how the receivers' decision is affected by the address forgery (spoofing) alone.

To this end, we configure 5 different types of email content for our study: (1) a blank email, (2) a blank email with a benign URL (`http://google.com`), (3) a blank email with a benign attachment (an empty text file). Then we have (4) a benign email with actual content. This email is a real-world legitimate email that informs a colleague about the change of time for a meeting. The reason for using "benign" content is to test how much the "spoofing" factor alone contributes to the email providers' decisions. In addition, to test whether a phishing email can penetrate the target service, we also include (5) an email with phishing content. This phishing email is a real-world sample from a phishing attack targeting our institution recently. The email impersonates the technical support to notify the victim that her internal account has been suspended and ask her to re-activate the account using a URL (to an Amazon EC2 server).

Sender IP. The IP address of the sender's mail server may also affect the spoofing success. We configure a *static* IP address and a *dynamic* IP address. Typically, mail servers need to be hosted on a static IP. In practice, attackers may use dynamic IPs for the lower cost.

Email Client. We examine how different email clients warn users of forged emails. We consider 3 common email clients: (1) a web client, (2) a mobile app, and (3) a third-party email client. All the 35 selected services have a web interface, and 28 have a dedicated mobile app. Third-party clients refer to the email applications (*e.g.*, Microsoft Outlook and Apple Mail) that allow users to check emails from any email providers.

5 Spoofing Experiment Results

In this section, we describe the results of our experiments. First, to provide the context, we measure the authentication protocols that the target email providers use to detect forged emails. Then, we examine how email providers handle forged emails and identify the key factors in the decision making. For emails that reached the inbox, we examine whether and how email providers warn users about their potential risks. Note that in this section, the all experiment results reflect the state of the target email services as of January 2018.

5.1 Authentication Mechanisms

To better interpret the results, we first examine how the 35 email providers authenticate incoming emails. One way of knowing their authentication protocols is to analyze the email headers and look for SPF/DKIM/DMARC authentication results. However, not all the email providers add the authentication results to the header (*e.g.*, `qq.com`) Instead, we follow a more reliable method [27] by setting up an *authoritative* DNS server for our own domain and sending an email from our domain. In the meantime, the authoritative DNS server will wait and see whether the target email service will query the SPF/DKIM/DMARC record. We set the TTL of the SPF, DKIM and DMARC records as 1 (second) to force the target email service always querying our *authoritative* DNS server. The results are shown in Table 2 (left 4 columns). 35 email providers can be grouped into 3 categories based on their protocols:

- **Full Authentication (16):** Email services that perform all three authentication checks (SPF, DKIM and DMARC). This category includes the most popular email services such as Gmail, Hotmail and iCloud.
- **SPF/DKIM but no DMARC (15):** Email services that check either SPF/DKIM, but do not check the sender's DMARC policy. These email services are likely to make decisions on their own.
- **No Authentication (4):** Email services that do not perform any of the three authentication protocols.

5.2 Decisions on Forged Emails

Next, we examine the decision-making process on forged emails. For each of the 35 target email services, we test all the possible combinations of the parameter settings (30 spoofed addresses $\times$ 5 types of email content $\times$ 2 IP

Email Provider	Supported Protocols			Overall Rate	IP		Spoofed Address Profile			Email Content				
	SPF	DKIM	DMARC	n=1500	Static 750	Dynamic 750	None 500	Related 500	Strict 500	BLK 300	URL 300	Atta. 300	Benign 300	Phish. 300
mail.ru	✓	✓	✓	0.69	0.69	0.69	1.00	0.99	0.07	0.70	0.69	0.69	0.68	0.68
fastmail.com	✓	✓	✓	0.66	1.00	0.32	0.70	0.65	0.64	0.67	0.66	0.67	0.67	0.65
163.com	✓	✓	✓	0.58	0.66	0.50	0.73	0.54	0.47	0.53	0.60	0.45	0.66	0.66
126.com	✓	✓	✓	0.57	0.66	0.48	0.74	0.54	0.43	0.54	0.56	0.46	0.65	0.64
gmail.com	✓	✓	✓	0.53	0.56	0.51	0.93	0.66	0.00	0.58	0.58	0.50	0.60	0.40
gmail inbox	✓	✓	✓	0.53	0.56	0.51	0.93	0.66	0.00	0.58	0.58	0.50	0.60	0.40
naver.com	✓	✓	✓	0.50	0.50	0.51	0.95	0.56	0.00	0.51	0.50	0.50	0.50	0.50
yeah.net	✓	✓	✓	0.36	0.51	0.21	0.44	0.38	0.26	0.23	0.35	0.34	0.61	0.28
tutanota.com	✓	✓	✓	0.36	0.41	0.30	0.90	0.17	0.00	0.39	0.39	0.20	0.39	0.39
yahoo.com	✓	✓	✓	0.35	0.67	0.03	0.52	0.52	0.00	0.33	0.34	0.33	0.38	0.35
inbox.lv	✓	✓	✓	0.32	0.63	0.00	0.50	0.45	0.00	0.32	0.32	0.32	0.32	0.32
protonmail.com	✓	✓	✓	0.30	0.60	0.00	0.45	0.45	0.00	0.32	0.26	0.29	0.31	0.32
seznam.cz	✓	✓	✓	0.24	0.48	0.00	0.35	0.25	0.13	0.35	0.35	0.35	0.08	0.08
aol.com	✓	✓	✓	0.18	0.16	0.19	0.29	0.25	0.00	0.24	0.20	0.22	0.23	0.00
icloud.com	✓	✓	✓	0.07	0.10	0.04	0.11	0.09	0.00	0.01	0.01	0.01	0.17	0.14
hotmail.com	✓	✓	✓	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
juno.com	✓	✓	×	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
sina.com	✓	✓	×	0.79	0.79	0.79	1.00	0.60	0.76	0.80	0.79	0.78	0.79	0.78
op.pl	✓	✓	×	0.71	0.71	0.71	1.00	0.72	0.40	0.71	0.71	0.71	0.71	0.71
sapo.pt	✓	×	×	0.59	0.67	0.50	0.91	0.54	0.31	0.64	0.53	0.49	0.63	0.64
zoho.com	✓	✓	×	0.58	0.57	0.58	0.99	0.54	0.21	0.59	0.54	0.59	0.59	0.59
qq.com	✓	✓	×	0.43	0.80	0.06	0.57	0.42	0.29	0.43	0.44	0.43	0.41	0.43
mynet.com	✓	✓	×	0.35	0.63	0.07	0.04	0.28	0.37	0.47	0.35	0.07	0.43	0.43
gmx.com	✓	✓	×	0.27	0.54	0.00	0.38	0.27	0.17	0.30	0.06	0.30	0.35	0.35
mail.com	✓	✓	×	0.27	0.54	0.00	0.37	0.27	0.17	0.29	0.06	0.30	0.35	0.35
daum.net	✓	×	×	0.27	0.52	0.01	0.33	0.29	0.18	0.28	0.26	0.27	0.27	0.25
runbox.com	✓	✓	×	0.24	0.48	0.00	0.28	0.26	0.19	0.25	0.00	0.00	0.48	0.48
interia.pl	✓	×	×	0.14	0.28	0.00	0.20	0.14	0.08	0.01	0.00	0.00	0.36	0.34
o2.pl	✓	✓	×	0.12	0.20	0.04	0.22	0.12	0.02	0.23	0.03	0.23	0.07	0.03
wp.pl	✓	✓	×	0.11	0.20	0.04	0.20	0.12	0.02	0.23	0.03	0.23	0.04	0.03
sohu.com	✓	×	×	0.03	0.03	0.03	0.02	0.03	0.03	0.04	0.04	0.01	0.03	0.03
t-online.de	×	×	×	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
excite.com	×	×	×	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
freemail.hu	×	×	×	0.99	0.99	0.99	1.00	1.00	0.96	0.97	1.00	0.97	1.00	1.00
rediffmail.com	×	×	×	0.78	0.79	0.78	0.74	0.80	0.80	0.76	0.79	0.76	0.79	0.79

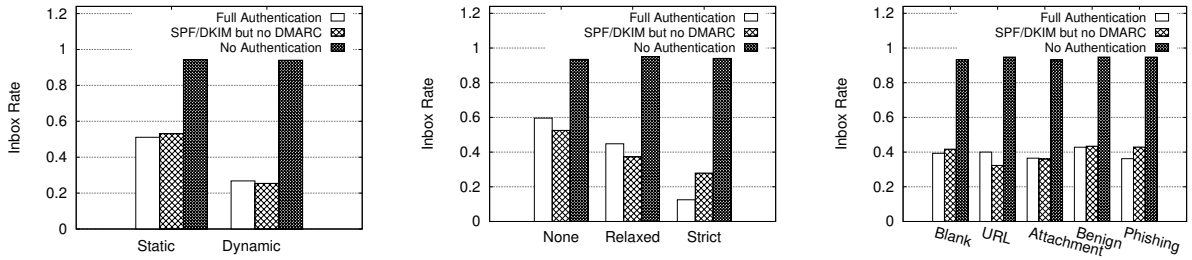
Table 2: The ratio of emails that reached the inbox (inbox rate). We break down the inbox rate for emails with different configuration parameters (sender IP, the SPF/DKIM/DMARC profile of the sender address, and the email content).

addresses), and then repeat the experiments for 5 times. Each email service receives 300 × 5 = 1,500 emails (52,500 emails in total). We shuffled all the emails and send them in randomized orders. We also set a sending time interval of 10 minutes (per email service) to minimize the impact to the target mail server. The experiment was conducted in December 2017– January 2018. Note the volume of emails in the experiment is considered very low compared to the *hundreds of billions* of emails sent over the Internet every day [5]. We intentionally limit our experiment scale so that the experiment emails would not impact the target services (and their email filters) in any significant ways. The randomized order and the slow sending speed helps to reduce the impact of the earlier emails to the later ones in the experiments.

After the experiment, we rely on IMAP/POP to retrieve the emails from the target email provider. For a few providers that do not support IMAP or POP, we use a browser-based crawler to retrieve the emails directly through the web interface. As shown in Table 2, we group email providers based on the supported authentication protocols. Within each group, we rank email providers based on the *inbox rate*, which is the ratio of emails that arrived the inbox over the total number of emails sent. Emails that did not arrive the inbox were either placed in the spam folder or completely blocked by the email providers.

Ratio of Emails in the Inbox. Table 2 shows that the vast majority of email services can be successfully penetrated. 34 out of the 35 email services allowed at least one forged email to arrive the inbox. The only exception is Hotmail which blocked all the forged emails. 33 out of 35 services allowed at least one *phishing* email to get into the inbox. In particular, the phishing email has penetrated email providers that perform full authentications (*e.g.*, Gmail, iCloud, Yahoo Mail) when spoofing sender domains that do not have a strict reject DMARC policy. In addition, providers such as `juno.com`, `t-online.de`, and `excite.com` did not block forged emails at all with a 100% inbox rate. `juno.com` actually checked both SPF and DKIM. This suggests that even though the email providers might have detected the email forgery, they still deliver the email to the user inbox.

Impact of Receiver's Authentication. Table 2 shows that email providers' authentication methods affect the spoofing result. For email providers that perform no authentication, the aggregated inbox rate is 94.2%. In comparison, the aggregated inbox rate is much lower for email providers that perform a full authentication



(a) Sender IP (b) Sender Address SPF/DK./DMA. Profile (c) Email Content

Figure 5: The aggregated rato of emails that reached the user inbox (inbox rate). The legend displays the 3 authentication groups of the receivers. Each subfigure shows the breakdown results for emails with specific configurations.

(39.0%) and email providers that just perform SPF/DKIM (39.3%). To examine the statistical significance of the differences, we apply Chi-Squared test on emails sent to the three types of email providers. The result confirms that emails are more likely to reach the inbox of "no-authentication" providers compared to the two other groups with statistical significance (both $p < 0.01$).

However, the difference between the "full-authentication" email providers and the "SPF/DKIM only" email providers are *not* statistically significant ($p = 0.495$). This indicates that the DMARC check has a relatively minor effect. Table 2 shows that DMARC check primarily affects emails where the spoofed domain has a "strict" reject policy. However, even with a full-authentication, the inbox rate of these emails is not always 0.00 (*e.g.*, `mail.ru`, `fastmail.com`, `163.com`, `126.com`, `yeah.net`, `seznam.cz`). This is because certain email providers would consider the DMARC policy as a "suggested action", but do not always enforce the policy.

Impact of the Sender IP. To better illustrate the impact of different email configurations, we plot Figure 5. We first group the target email providers based on their authentication method (3 groups), and then calculate the *aggregated inbox rate* for a specific configuration setting. As shown in Figure 5(a), emails that sent from a static IP has a higher chance to reach the inbox (56.9%) compared to those from a dynamic IP (33.9%). Chi-Square statistical analysis shows the difference is statistically significant ($p < 0.0001$). In practice, however, dynamic IPs are still a viable option for attackers since they are cheaper.

To ensure the validity of results, we have performed additional analysis to make sure our IPs were not blacklisted during the experiment. More specifically, we analyze our experiment traces to monitor the inbox rate throughout the experiment process. In our experiment, each email service receives 1500 emails, and we checked the inbox rate per 100 emails over time. If our IPs were blacklisted during the experiment, there should be a sharp decrease in the inbox rate at some point. We did not observe that in any of the tested email services. We also checked 94 public blacklists [2], and our IPs are not on any of them.

Impact of Spoofed Sender Domain. Figure 5(b) demonstrates the impact of the spoofed sender address. Overall, spoofing a sender domain that has no SPF/DKIM/DMARC records yields a higher inbox rate (60.5%). Spoofing a sender domain with SPF/DKIM and a "relaxed" failing policy has a lower inbox rate (47.3%). Not too surprisingly, domains with SPF/DKIM records and a "strict" reject policy is the most difficult to spoof (inbox rate of 28.4%). Chi-Square statistical analysis shows the differences are significant ($p < 0.00001$). The result confirms the benefits of publishing SPF/DKIM/DMARC records. However, publishing these records cannot completely prevent being spoofed, since email providers may still deliver emails that failed the SPF/DKIM authentication.

Impact of Email Content. Figure 5(c) shows that the inbox rates are not very different for different email content. The differences are small but not by chance (Chi-Squared test $p < 0.00001$). This suggests that our result is not dependent on a specific email content chosen for the study. Recall that we specifically use benign-looking content to minimize the impact of spam filters, so that we can test how much the "spoofing" factor contributes to email providers' decisions. This does not mean that email content has no impact on the decision making. On the contrary, if an email has a blacklisted URL or a known malware as the attachment, we expected more emails will be blocked (which is not our study purpose). Our result simply shows that today's attackers can easily apply spoofing to conduct targeted spear phishing. In the context of spear phishing, it is a reasonable assumption that the attacker will craft benign-looking content with URLs that have not been blacklisted yet [33].

Ranking the Factors. To determine which factors contribute more to a successful penetration, we perform

[2] https://mxtoolbox.com/blacklists.aspx

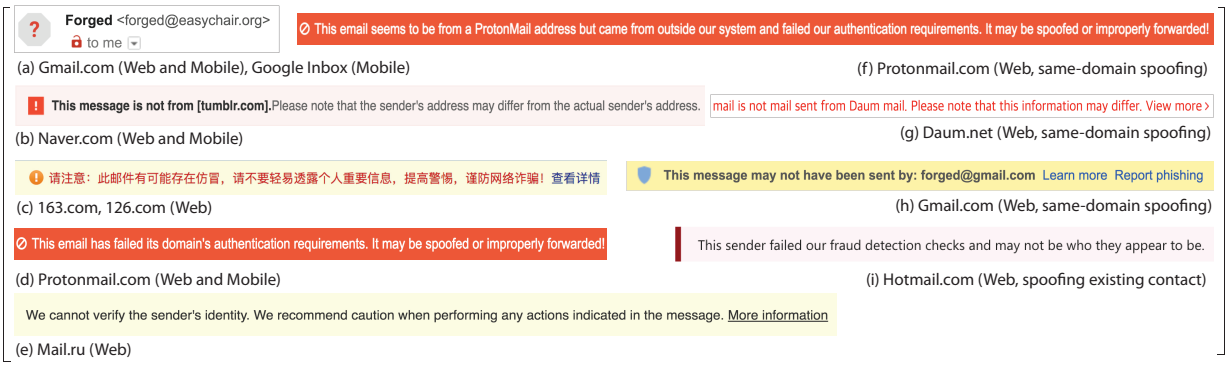


Figure 6: Security indicators on forged emails from 9 email providers. (a)–(e) are for regular forged emails. (f)–(h) only show up when the spoofed sender and the receiver belong to the same provider. (i) only shows up when spoofing an existing contact.

Feature	Chi^2	Mutual Info
Receiver authentication method	6497.93	0.0707
Spoofed sender address	3658.72	0.0356
Sender IP	2799.51	0.0269
Email content	115.27	0.0011

Table 3: Feature ranking.

a "feature ranking" analysis. We divide all the emails into two classes: *positive* (inbox) and *negative* (spam folder or blocked). For each email, we calculate four features: email content (F_1), sender address profile (F_2), receiver authentication group (F_3), and sender IP (F_4), all of which are categorical variables. Then we rank features based on their distinguishing power to classify emails into the two classes using standard metrics: Chi-Square Statistics [45] and Mutual Information [17]. As shown in Table 3, consistently, "receiver authentication method" is the most important factor, followed by the "spoofed sender address". Note that this analysis only compares the relative importance of factors in our experiment. We are not trying to reverse-engineer the complete defense system, which requires analyzing more features.

Discussion. It takes both the sender and the receiver to make a reliable email authentication. When one of them fails to do their job, there is a higher chance for the forged email to reach the inbox. In addition, email providers tend to prioritize email delivery over security. When an email fails the authentication, most email providers (including Gmail and iCloud) would still deliver the email as long as the policy of the spoofed domain is not "reject". Based on the earlier measurement result (§3), only 13% of the 1 million domains have set a "reject" or "hard fail" policy, which leaves plenty of room for attackers to perform spoofing.

Our analysis also revealed a vulnerability in two email services (`sapo.p` and `runbox.com`), which would allow an attacker to send spoofing emails through the email provider's IP. Since this is a different threat model, we discuss the details of this vulnerability in Appendix B.

5.3 Email Clients and Security Indicators

For emails that reached the user inbox, we next examine the security indicators on email interfaces to warn users. Again the results represent the state of email services as of January 2018.

Web Client. We find that only 6 email services have displayed security indicators on forged emails including `Gmail`, and `protonmail`, `naver`, `mail.ru`, `163.com` and `126.com` (Figure 6 (a)–(e)). Other email services display forged emails without any visual alert (*e.g.*, Yahoo Mail, iCloud). Particularly, Gmail and Google Inbox are from the same company, but the web version of Google Inbox has no security indicator. Gmail's indicator is a "question mark" on the sender's icon. Only when users move the mouse over the image, it will show the following message: *"Gmail could not verify that <sender> actually sent this message (and not a spammer)"*. The red lock icon is not related to spoofing, but to indicate the communication between MX servers is unencrypted. On the other hand, services like `naver`, `163.com` and `protonmail` use explicit text messages to warn users.

Mobile Client. Even fewer mobile email apps have adopted security indicators. Out of the 28 email services with a dedicated mobile app, only 4 services have mobile security indicators including `naver`, `protonmail`, `Gmail`, and `google inbox`. The other services removed the security indicators for mobile users. Compared to the web interface, mobile apps have very limited screen size. Developers often remove "less important" information to keep a clean interface. Unfortunately, the security indicators are among the removed elements.

Misleading UI	Email Providers (25 out of 35)
Sender Photo (6)	G-Inbox, Gmail, zoho, icloud*, gmx†, mail.com†
Name Card (17)	yahoo, hotmail, tutanota, seznam.cz, fastmail, gmx, mail.com, Gmail*, sina*, juno*, aol*, 163.com†, 126.com†, yeah.net†, sohu†, naver†, zoho†
Email History (17)	hotmail, 163.com, 126.com, yeah.net, qq, zoho, mail.ru, yahoo*, Gmail*, sina*, naver*, op.pl*, interia.pl*, daum.net* gmx.com*, mail*, inbox.lv*

Table 4: Misleading UI elements when the attacker spoofs an existing contact. (∗) indicates web interface only. (†) indicates mobile only.

Third-party Client. Finally, we check emails using third-party clients including Microsoft Outlook, Apple Mail, and Yahoo Web Mail. We test both desktop and mobile versions, and find that *none* of them provide security indicators for the forged emails.

5.4 Misleading UI Elements

We find that attackers can trigger misleading UI elements to make the forged email look realistic.

Spoofing an Existing Contact. When an attacker spoofs an existing contact of the receiver, the forged email can automatically load misleading UI elements such as the contact's photo, name card, or previous email conversations. We perform a quick experiment as follows: First, we create an "existing contact" (`contact@vt.edu`) for each receiver account in the 35 email services, and add a name, a profile photo and a phone number (if allowed). Then we spoof this contact's address (`contact@vt.edu`) to send forged emails. Table 4 shows the 25 email providers that have misleading UIs. Example screenshots are shown in Appendix C. We believe that these designs aim to improve the usability of the email service by providing the context for the sender. However, when the sender address is actually spoofed, these UI elements would help attackers to make the forged email look more authentic.

In addition, spoofing an existing contact allows forged emails to penetrate new email providers. For example, Hotmail blocked *all* the forged emails in Table 2. However, when we spoof an existing contact, Hotmail delivers the forged email to the inbox and adds a special warning sign as shown in Figure 6(i).

Same-domain Spoofing. Another way to trigger the misleading UI element is to spoof an email address that belongs to the same email provider as the receiver. For example, when spoofing `<forged@seznam.cz>` to send an email to `<test@eznam.cz>`, the profile photo of the spoofed sender will be automatically loaded. Since

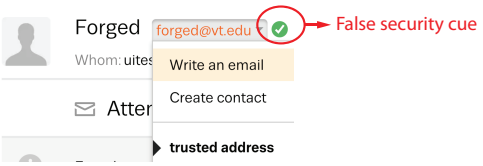


Figure 7: Seznam.cz displays a "trusted address" sign on a forged address.

the spoofed sender is from the same email provider, the email provider can directly load the sender's photo from its own database. This phenomenon applies to Google Inbox and Gmail (mobile) too. However, email providers also alert users with special security indicators. As shown in Figure 6(f)-(h), related email providers include `protonmail`, `Gmail` and `daum.net`. Together with previously observed security indicators, there are in total 9 email providers that provide at least one type of security indicators.

False Security Indicators. One email provider `seznam.cz` displays a false security indicator to users. `seznam.cz` performs full authentications but still delivers spoofed emails to the inbox. Figure 7 shows that `seznam.cz` displays a green checkmark on the sender address even though the address is forged. When users click on the icon, it displays "trusted address", which is likely to give users a false sense of security.

6 Effectiveness of Security Indicators

As an end-to-end study, we next examine the last hop — how users react to spoofing emails. Our result so far shows that a few email providers have implemented visual security indicators on the email interface to warn users of the forged emails. In the following, we seek to understand how effective these security indicators are to improve user efficacy in detecting spoofed phishing emails.

6.1 Experiment Methodology

To evaluate the effectiveness of security indicators, we design an experiment where participants receive a phishing email with a forged sender address. By controlling the security indicators on the interface, we assess how well security indicators help users to handle phishing emails securely.

Implementing this idea faces a key challenge, which is to capture the realistic user reactions to the email. Ideally, participants should examine the phishing email *without knowing that they are in an experiment*. However, this leads to practical difficulties to set up the user study and obtain the informed user consent up front. To

this end, we introduce *deception* to the study methodology. At the high level, we use a distractive task to hide the true purpose of the study *before* and *during* the study. Then *after* the study is completed, we debrief the users to obtain the informed consent. Working closely with our IRB, we have followed the ethical practices to conduct the phishing test.

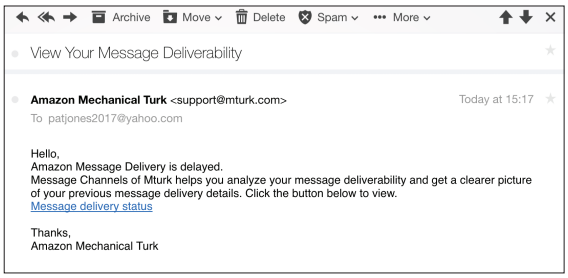
Procedure. We frame the study as a survey to understand users' email habits. The true purpose is hidden from the participants. This study contains two phases. Phase1 is to set up the deception and phase 2 carries out the phishing experiment.

Phase1: The participants start by entering their *own email addresses*. Then we immediately send the participants an email and instruct the participants to check this email from their email accounts. The email contains a tracking pixel (a 1×1 transparent image) to measure if the email has been opened. After that, we ask a few questions about the email (to make sure they actually opened the email). Then we ask other distractive survey questions about their email usage habits. *Phase1* has three purposes: (1) to make sure the participants actually own the email address; (2) to test if the tracking pixel works, considering some users may configure their email service to block images and HTML; (3) to set up the deception. After phase1, we give the participants the impression that the survey is completed (participants get paid after *phase1*). In this way, participants would not expect the second phishing email.

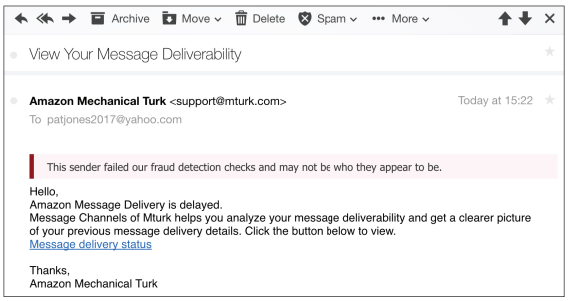
Phase2: We wait for 10 days and send the phishing email. The phishing email contains a benign URL pointing to our own server to measure whether the URL is clicked. In addition, the email body contains a tracking pixel to measure if the email has been opened. As shown in Figure 8, we impersonate the tech-support of Amazon Mechanical Turk (`support@mturk.com`) to send the phishing email that informs some technical problems. This email actually targeted our own institution before. The phishing email is only sent to users whose email service is not configured to block HTML or tracking pixels (based on *phase1*).

We wait for another 20 days to monitor user clicks. After the study, we send a debriefing email which explains the true purpose of the experiment and obtains the informed consent. Participants can withdraw their data anytime. By the time of our submission, none of the users have requested to withdraw their data.

Security Indicators. Based on our previous measurement results, most email services adopted text-based indicators (Figure 6(b)-(i)). Even Gmail's special indicator (Figure 6(a)) will display a text message when users move the mouse over. To this end, we use the text-based indicator and make two settings, namely *with security indicator* and *without security indicator*. For the group *without security indicator*, we recruit users from Yahoo Mail. We choose Yahoo Mail users because Yahoo Mail is the largest email service that has not implemented any security indicators. For the comparison group *with security indicator*, we still recruit Yahoo Mail users for consistency, and add our own security indicators to the interface. More specifically, when sending emails, we can embed a piece of HTML code in the email body to display a text-based indicator. This is exactly how most email providers insert their visual indicators in the email body (except for Gmail).



(a) Without Security Indicator



(b) With Security Indicator

Figure 8: The phishing email screenshot.

In *phase2*, we cannot control if a user would use the mobile app or the website to read the email. This is not a big issue for Yahoo Mail users. Yahoo's web and mobile clients both render HTML by default. The text-based indicator is embedded in the email body by us, which will be displayed consistently for both web and mobile users (confirmed by our own tests).

Recruiting Participants. To collect enough data points from *phase 2*, we need to recruit a large number of users given that many users may not open our email. We choose Amazon Mechanical Turk (MTurk), the most popular crowdsourcing platform to recruit participants. MTurk users are slightly more diverse than other Internet samples as well as college student samples. Using Amazon Mechanical Turk may introduce biases in terms of the user populations. However, the diversity is reportedly better than surveying the university students [9]. To avoid non-serious users, we apply the screening criteria

Phase	Users	w/o Indict.	w/ Indict.
Phase1	All Participants	243	245
	Not Block Pixel	176	179
Phase2	Opened Email	94	86
	Clicked URL	46	32
Click Rate	Overall	26.1%	17.9%
	After Open Email	48.9%	37.2%

Table 5: User study statistics.

Users	w/o Indicator		w/ Indicator	
	Desktop	Mobile	Desktop	Mobile
Opened Email	45	49	41	45
Clicked URL	21	25	15	17
Click Rate	46.7%	51.0%	36.6%	37.8%

Table 6: User study statistics for different user-agents.

that are commonly used in MTurk [10, 28]. We recruit users from the U.S. who have a minimum Human Intelligence Task (HIT) approval rate of 90%, and more than 50 approved HITs.

In total, we recruited $N = 488$ users from MTurk: 243 users for the "without security indicator" setting, and another 245 users for the "with security indicator" setting. Each user can only participate in *one setting for only once* to receive \$0.5. In the recruiting letter, we explicitly informed the users that we need to collect their email address. This may introduce self-selection biases: we are likely to recruit people who are willing to share their email address with our research team. Despite the potential bias, that the resulting user demographics are quite diverse: 49% are male and 51% are female. Most participants are 30–39 years old (39.1%), followed by users under 29 (31.8%), above 50 (14.5%), and 40–49 (14.5%). Most of the participants have a bachelor degree (35.0%) or a college degree (33.8%), followed by those with a graduate degree (20.7%) and high-school graduates (10.5%).

Ethics Guidelines. Our study received IRB approval, and we have taken active steps to protect the participants. First, only benign URLs are placed in the emails which point to our own server. Clicking on the URL does not introduce practical risks to the participants or their computers. Although we can see the participant's IP, we choose not to store the IP information in our dataset. In addition, we followed the recommended practice from IRB to conduct the deceptive experiment. In the experiment instruction, we omit information only if it is absolutely necessary (*e.g.*, the purpose of the study and details about the second email). Revealing such information upfront will invalidate our results. After the experiment, we immediately contact the participants to explain our real purpose and the detailed procedure. We offer the opportunity for the participants to opt out. Users who opt-out still get the full payment.

6.2 Experiment Results

We analyze experiment results to answer the following questions. First, how effective are security indicators in protecting users? Second, how does the impact of security indicators vary across different user demographics?

Click-through Rate. Table 5 shows the statistics for the phishing results. For phase-2, we calculate two click-through rates. First, out of all the participants that *received* the phishing email, the click-through rate with security indicator is 32/179=17.9%. The click-through rate without security indicator is higher: 46/176=26.1%. However, this comparison is not entirely fair, because many users did not open the email, and thus did not even see the security indicator at all.

In order to examine the impact of the security indicator, we also calculate the click-through rate based on users who *opened* the email. More specifically, we sent phishing emails to the 176 and 179 users who did not block tracking pixels, and 94 and 86 of them have opened the email. This returns the email-opening rate of 53.4% and 48.9%. Among these users, the corresponding click-through rates are 48.9% (without security indicator) and 37.2% (with security indicator) respectively. The results indicate that security indicators have a positive impact to reduce risky user actions. When the security indicator is presented, the click rate is *numerically* lower compared to that without security indicators. The difference, however, is not very significant (Fisher's exact test $p = 0.1329$). We use Fisher's exact test instead of the Chi-square test due to the relatively small sample size. The result suggests that the security indicator has a moderately positive impact.

User Agents. In our experiment, we have recorded the "User-Agent" when the user opens the email, which helps to infer the type of device that a user was using to check the email. Recall that no matter what device the user was using, our security indicator (embedded in the email body) will show up regardless. Table 6 shows that mobile users are more likely to click on the phishing link compared with desktop users, but the difference is not significant.

Demographic Factors. In Figure 9, we cross-examine the results with respect to the demographic factors. To make sure each demographic group contains enough users, we create binary groups for each factor. For "education level", we divide users into High-Edu (bachelor degree or higher) and Low-Edu (no bachelor degree). For "age", we divide users into Young (age<40)

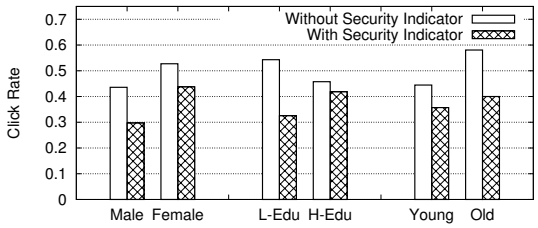


Figure 9: The joint impact of demographic factors and security indicators on click rates.

and Old (age>=40). The thresholds are chosen so that the two groups are of relatively even sizes. As shown in Figure 9, the click rates are consistently lower when a security indicator is presented for all the demographic groups. The differences are still insignificant. Fisher's exact test shows that the smallest $p = 0.06$, which is produced by the *low-edu* group. Overall, our result confirms the positive impact of the security indicator across different user demographics, and also suggests the impact is limited. The security indicator alone is not enough to mitigate the risk.

7 Discussion

In this section, we summarize our results and discuss their implications for defending against email spoofing and broadly spear phishing attacks. In addition, we discuss the new changes made by the email services after our experiment, and our future research directions.

7.1 Implications of Our Results

Email Availability vs. Security. Our study shows many email providers choose to deliver a forged email to the inbox even when the email fails the authentication. This is a difficult trade-off between security and email availability. If an email provider blocks all the unverified emails, users are likely to lose their emails (*e.g.*, from domains that did not publish an SPF, DKIM or DMARC record). Losing legitimate emails is unacceptable for email services which will easily drive users away.

The challenge is to accelerate the adoption of SPF, DKIM and DMARC. Despite the efforts of the Internet Engineering Task Force (IETF), these protocols still have limitations to handle special email scenarios such as mail forwarding and mailing lists, creating further obstacles to a wide adoption [40, 19, 37]. Our measurement shows a low adoption rate of SPF (44.9%) and DMARC (5.1%) among the Internet hosts. From the email provider's perspective, the ratio of unverified inbound emails is likely to be lower since heavy email-sending domains are likely to adopt these protocols. According to the statistics from Google in 2015 [23], most inbound emails to Gmails have either SPF (92%) or DKIM (83.0%), but only a small portion (26.1%) has a DMARC policy. This presents an on-going challenge since spear phishing doesn't require a large volume of emails to get in. Sometimes one email is sufficient to breach a target network.

Countermeasures and Suggestions. First and foremost, email providers should consider adopting SPF, DKIM and DMARC. Even though they cannot authenticate all the incoming emails, these protocols allow the email providers to make more informed decisions. Further research is needed to ease the deployment process and help to avoid disruptions to the existing email operations [15].

In addition, if the email providers decide to deliver an unverified email to the inbox, we believe it is necessary to place a security indicator to warn users based on our user study results. A potential benefit is that the security indicator can act as a forcing function for sender domains to configure their SPF/DKIM/DMARC correctly.

Third, we argue that email providers should make the security indicators *consistently* for different interfaces. Currently, mobile users are exposed to a higher-level of risks due to the lack of security indicators. Another example is that Google Inbox (web) users are less protected compared to users that use Gmail's interface.

Finally, the misleading UI elements such as "profile photo" and "email history" should be disabled for emails with unverified sender addresses. This should apply to both spoofing an existing contact and spoofing users in of same email provider. So far, we have communicated our results with the Gmail team and provided the suggestions on improving the current security indicators. We are in the process of communicating with other email providers covered in our study.

New Protocols BIMI and ARC. Recently, new protocols are developed to enhance spoofing detection. For example, BIMI (Brand Indicators for Message Identification) is a protocol built on DMARC. After confirming the authenticity of the email sender via DMARC, the email client can display a BIMI logo as a security indicator for the sender brand. This means emails with a BIMI logo are verified, but those without the BIMI logo are not necessarily malicious.

ARC (Authenticated Received Chain) is an underdevelopment protocol that works on top of SPF, DKIM and DMARC. ARC aims to address the problems caused by mail forwarding and mailing lists. For example, when an email is sent through a mailing list, the email sending IP and the email content might be changed (*e.g.*, adding a footer) which will break SPF or DKIM. ARC proposes to preserve the email authentication results through differ-

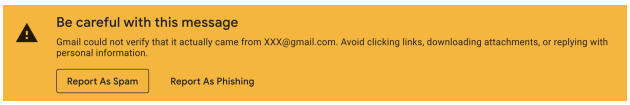


Figure 10: Gmail's new warning message for same-domain spoofing.

ent sending scenarios. For both ARC and BIMI, they are likely to face the same challenge to be widely adopted just like DMARC (standardized in 2015).

7.2 UI Updates from Email Services

A few email services have updated their user interfaces during January – June in 2018. Particularly, after we communicate our results to the Gmail team, we notice some major improvements. First, when we perform the same-domain spoofing (*i.e.*, spoofing a Gmail address), in addition to the question-mark sign, there is a new warning message added to the email body as shown in Figure 10. Second, the new mobile Gmail app no longer displays the "misleading" profile photos on unverified messages (regardless spoofing existing contact or the same-domain account). The same changes are applied to the new Google Inbox app too. However, the mobile clients are still not as informative as the web version. For example, there is no explanation message on the question-mark sign on the mobile apps. In addition, the new warning message (Figure 10) has not been consistently added to the mobile apps either.

`Inbox.lv` has launched its mobile app recently. Like its web version, the mobile app does not provide a security indicator. However, the UI of the mobile app is simplified which no longer loads misleading elements (*e.g.*, profile photos) for unverified emails. Yahoo Mail and Zoho also updated their web interfaces but the updates were not related to security features.

7.3 Open Questions & Limitations

Open Questions. It is unlikely that the email spoofing problem can quickly go away given the slow adoption rate of the authentication protocols. Further research is needed to design more effective indicators to maximize its impact on users. Another related question is how to maintain the long-term effectiveness of security indicators and overcome the "warning fatigue" [8]. Finally, user training/education will be needed to teach users how to interpret the warning message, and handle questionable emails securely. For security-critical users (*e.g.*, journalists, government agents, military personnel), an alternative approach is to use PGP to prevent email spoofing [29]. Extensive work is still needed to make PGP widely accessible and usable for the broad Internet population [30, 48].

Study Limitations. Our study has a few limitations. First, our measurement only covers public email services. Future work will explore if the conclusion also applies to non-public email services. Second, while we have taken significant efforts to maintain the validity of the phishing test, there are still limits to what we can control. For ethical considerations, we cannot fully scale-up the experiments beyond the 488 users, which limited the number of variables that we can test. Our experiment only tested a binary condition (with or without a security indicator) on one email content. Future work is needed to cover more variables to explore the design space such as the wording of the warning messages, the color and the font of the security indicator, the phishing email content, and the user population (*e.g.*, beyond the MTurk and Yahoo Mail users). Finally, we use "clicking on the phishing URL" as a measure of risky actions, which is still not the final step of a phishing attack. However, tricking users to give way their actual passwords would have a major ethical implication, and we decided not to pursue this step.

8 Related Work

Email Confidentiality, Integrity and Authenticity. SMTP extensions such as SPF, DKIM, DMARC and STARTTLS are used to provide security properties for email transport. Recently, researchers conducted detailed measurements on the *server-side* usage of these protocols [23, 27, 34, 36]. Unlike prior work, our work shows an end-to-end view and demonstrate the gaps between server-side spoofing detection and the user-end notifications. Our study is complementary to existing work to depict a more complete picture.

Email Phishing. Prior works have developed phishing detection methods based on features extracted from email content and headers [20, 22, 26, 35, 51, 57]. Phishing detection is different from spam filtering [58] because phishing emails are not necessarily sent in bulks [65] but can be highly targeted [33]. Other than spoofing, attackers may also apply typosquatting or unicode characters [6] to make the sender address *appear similar* (but not identical) to what they want to impersonate. Such sender address is a strong indicator of phishing which has been used to detect phishing emails [42, 44]. Another line of research focuses on the *phishing website*, which is usually the landing page of the URL in a phishing email [18, 32, 63, 68, 71, 72].

Human factors (demographics, personality, cognitive biases, fatigue) would affect users response to phishing [52, 31, 38, 53, 60, 64, 66, 69, 16, 47]. The

study results have been used to facilitate phishing training [67]. While most of these studies use the "role-playing" method, where users read phishing emails in the simulated setting. There are rare exceptions [38, 52] where the researchers conducted a real-world phishing experiment. Researchers have demonstrated the behavioral differences in the role-playing experiments with reality [59]. Our work is the first to examine the impact of security indicators on phishing emails using realistic phishing tests.

Visual Security Indicators. Security Indicators are commonly used in web or mobile browsers to warn users of unencrypted web sessions [25, 39, 61, 49], phishing web pages [21, 24, 69, 70], and malware sites [7]. Existing work shows that users often ignore the security indicators due to a lack of understanding of the attack [69] or the frequent exposure to false alarms [43]. Researchers have explored various methods to make security UIs harder to ignore such as using attractors [13, 12, 14]. Our work is the first to measure the usage and effectiveness of security indicators on forged emails.

9 Conclusion

Through extensive end-to-end measurements and real-world phishing tests, our work reveals a concerning gap between the server-side spoofing detection and the actual protection on users. We demonstrate that most email providers allow forged emails to get to user inbox, while lacking the necessary warning mechanism to notify users (particularly on mobile apps). For the few email services that implemented security indicators, we show that security indicators have a positive impact on reducing risky user actions under phishing attacks but cannot eliminate the risk. We hope the results can help to draw more community attention to promoting the adoption of SMTP security extensions, and developing effective security indicators for the web and mobile email interfaces.

Acknowledgments

We would like to thank the anonymous reviewers for their helpful feedback. This project was supported in part by NSF grants CNS-1750101 and CNS-1717028. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of any funding agencies.

References

[1] Alexa. http://www.alexa.com.

[2] Phishing activity trends report, 1st 3rd quarters 2015. http://docs.apwg.org/reports/apwg_trends_report_q1-q3_2015.pdf.

[3] Postfix. http://www.postfix.org.

[4] Data breach investigations report. Verizon Inc., 2017. http://www.verizonenterprise.com/verizon-insights-lab/dbir/2017/.

[5] Email statistics report. The Radicati Group, 2017. http://www.radicati.com/wp/wp-content/uploads/2017/01/Email-Statistics-Report-2017-2021-Executive-Summary.pdf.

[6] AGTEN, P., JOOSEN, W., PIESSENS, F., AND NIKIFORAKIS, N. Seven months' worth of mistakes: A longitudinal study of typosquatting abuse. In *Proc. of NDSS* (2015).

[7] AKHAWE, D., AND FELT, A. P. Alice in warningland: A large-scale field study of browser security warning effectiveness. In *Proc. of USENIX Security* (2013).

[8] ANDERSON, B. B., VANCE, T., KIRWAN, C. B., EARGLE, D., AND HOWARD, S. Users aren't (necessarily) lazy: Using neurois to explain habituation to security warnings. In *Proc. of ICIS* (2014).

[9] ANTIN, J., AND SHAW, A. Social desirability bias and self-reports of motivation: A study of amazon mechanical turk in the us and india. In *Proc. of CHI* (2012).

[10] BILOGREVIC, I., HUGUENIN, K., MIHAILA, S., SHOKRI, R., AND HUBAUX, J.-P. Predicting users' motivations behind location check-ins and utility implications of privacy protection mechanisms. In *Proc. of NDSS* (2015).

[11] BLANZIERI, E., AND BRYL, A. A survey of learning-based techniques of email spam filtering. *Artificial Intelligence Review 29*, 1 (2008), 63–92.

[12] BRAVO-LILLO, C., CRANOR, L., AND KOMANDURI, S. Harder to ignore? revisiting pop-up fatigue and approaches to prevent it. In *Proc. of SOUPS* (2014).

[13] BRAVO-LILLO, C., CRANOR, L. F., DOWNS, J., AND KOMANDURI, S. Bridging the gap in computer security warnings: A mental model approach. In *Proc. of IEEE S&P* (2011).

[14] BRAVO-LILLO, C., KOMANDURI, S., CRANOR, L. F., REEDER, R. W., SLEEPER, M., DOWNS, J., AND SCHECHTER, S. Your attention please: Designing security-decision uis to make genuine risks harder to ignore. In *Proc. of SOUPS* (2013).

[15] CONSTANTIN, L. Yahoo email anti-spoofing policy breaks mailing lists. PC World, 2014. https://www.pcworld.com/article/2141120/yahoo-email-antispoofing-policy-breaks-mailing-lists.html.

[16] CONWAY, D., TAIB, R., HARRIS, M., YU, K., BERKOVSKY, S., AND CHEN, F. A qualitative investigation of bank employee experiences of information security and phishing. In *Proc. of SOUPS* (2017).

[17] COVER, T. M., AND THOMAS, J. A. *Elements of information theory*. John Wiley & Sons, 2012.

[18] CUI, Q., JOURDAN, G.-V., BOCHMANN, G. V., COUTURIER, R., AND ONUT, I.-V. Tracking phishing attacks over time. In *Proc. of WWW* (2017).

[19] D. CROCKER, T. HANSEN, M. K. Domainkeys identified mail (dkim) signatures, 2011. `https://tools.ietf.org/html/rfc6376`.

[20] DEWAN, P., KASHYAP, A., AND KUMARAGURU, P. Analyzing social and stylometric features to identify spear phishing emails. In *Proc. of eCrime* (2014).

[21] DHAMIJA, R., TYGAR, J. D., AND HEARST, M. Why phishing works. In *Proc. of CHI* (2006).

[22] DUMAN, S., KALKAN-CAKMAKCI, K., EGELE, M., ROBERTSON, W. K., AND KIRDA, E. Emailprofiler: Spearphishing filtering with header and stylometric features of emails. In *Proc. of COMPSAC* (2016).

[23] DURUMERIC, Z., ADRIAN, D., MIRIAN, A., KASTEN, J., BURSZTEIN, E., LIDZBORSKI, N., THOMAS, K., ERANTI, V., BAILEY, M., AND HALDERMAN, J. A. Neither snow nor rain nor mitm: An empirical analysis of email delivery security. In *Proc. of IMC* (2015).

[24] EGELMAN, S., CRANOR, L. F., AND HONG, J. You've been warned: An empirical study of the effectiveness of web browser phishing warnings. In *Proc. of CHI* (2008).

[25] FELT, A. P., ET AL. Rethinking connection security indicators. In *Proc. of SOUPS* (2016).

[26] FETTE, I., SADEH, N., AND TOMASIC, A. Learning to detect phishing emails. In *Proc. of WWW* (2007).

[27] FOSTER, I. D., LARSON, J., MASICH, M., SNOEREN, A. C., SAVAGE, S., AND LEVCHENKO, K. Security by any other name: On the effectiveness of provider based email security. In *Proc. of CCS* (2015).

[28] GADIRAJU, U., KAWASE, R., DIETZE, S., AND DEMARTINI, G. Understanding malicious behavior in crowdsourcing platforms: The case of online surveys. In *Proc. of CHI* (2015).

[29] GARFINKEL, S. *PGP: Pretty Good Privacy*, 1st ed. O'Reilly & Associates, Inc., 1996.

[30] GAW, S., FELTEN, E. W., AND FERNANDEZ-KELLY, P. Secrecy, flagging, and paranoia: Adoption criteria in encrypted email. In *Proc. of CHI* (2006).

[31] GREITZER, F. L., STROZER, J. R., COHEN, S., MOORE, A. P., MUNDIE, D., AND COWLEY, J. Analysis of unintentional insider threats deriving from social engineering exploits. In *Proc. of IEEE S&P Workshops* (2014).

[32] HAN, X., KHEIR, N., AND BALZAROTTI, D. Phisheye: Live monitoring of sandboxed phishing kits. In *Proc. of CCS* (2016).

[33] HO, G., SHARMA, A., JAVED, M., PAXSON, V., AND WAGNER, D. Detecting credential spearphishing in enterprise settings. In *Proc. of USENIX Security* (2017).

[34] HOLZ, R., AMANN, J., MEHANI, O., WACHS, M., AND KAAFAR, M. A. Tls in the wild: An internet-wide analysis of tls-based protocols for electronic communication. In *Proc. of NDSS* (2016).

[35] HONG, J. The state of phishing attacks. *Communications of the ACM 55*, 1 (2012).

[36] HU, H., PENG, P., AND WANG, G. Towards the adoption of anti-spoofing protocols. *CoRR abs/1711.06654* (2017).

[37] HU, H., PENG, P., AND WANG, G. Towards understanding the adoption of anti-spoofing protocols in email systems. In *Proc. of SecDev* (2018).

[38] JAGATIC, T. N., JOHNSON, N. A., JAKOBSSON, M., AND MENCZER, F. Social phishing. *Communications of the ACM 50*, 10 (2007).

[39] JOEL WEINBERGER, A. P. F. A week to remember the impact of browser warning storage policies. In *Proc. of SOUPS* (2016).

[40] KITTERMAN, S. Sender policy framework (spf), 2014. `https://tools.ietf.org/html/rfc7208`.

[41] KOCIENIEWSKI, D. Adobe announces security breach. The New York Times, 2013. `http://www.nytimes.com/2013/10/04/technology/adobe-announces-security-breach.html`.

[42] KRAMMER, V. Phishing defense against idn address spoofing attacks. In *Proc. of PST* (2006).

[43] KROL, K., MOROZ, M., AND SASSE, M. A. Don't work. can't work? why it's time to rethink security warnings. In *Proc. of CRiSIS* (2012).

[44] KUMARAGURU, P., RHEE, Y., ACQUISTI, A., CRANOR, L. F., HONG, J., AND NUNGE, E. Protecting people from phishing: The design and evaluation of an embedded training email system. In *Proc. of CHI* (2007).

[45] LANCASTER, H. O., AND SENETA, E. *Chi-square distribution*. Wiley Online Library, 1969.

[46] LARDINOIS, F. Gmail now has more than 1b monthly active users. Tech Crunch, 2016. `https://techcrunch.com/2016/02/01/gmail-now-has-more-than-1b-monthly-active-users/`.

[47] LASTDRAGER, E., GALLARDO, I. C., HARTEL, P., AND JUNGER, M. How effective is anti-phishing training for children? In *Proc. of SOUPS* (2017).

[48] LUBAR, K., AND IMAGES, G. After 3 years, why gmail's end-to-end encryption is still vapor. Wired, 2017. `https://www.wired.com/2017/02/3-years-gmails-end-end-encryption-still-vapor/`.

[49] LUO, M., STAROV, O., HONARMAND, N., AND NIKIFORAKIS, N. Hindsight: Understanding the evolution of ui vulnerabilities in mobile browsers. In *Proc. of CCS* (2017).

[50] M. KUCHERAWY, E. Z. Domain-based message authentication, reporting, and conformance (dmarc), 2015. `https://tools.ietf.org/html/rfc7489`.

[51] MCGRATH, D. K., AND GUPTA, M. Behind phishing: An examination of phisher modi operandi. In *Proc. of LEET* (2008).

[52] OLIVEIRA, D., ROCHA, H., YANG, H., ELLIS, D., DOMMARAJU, S., MURADOGLU, M., WEIR, D., SOLIMAN, A., LIN, T., AND EBNER, N. Dissecting spear phishing emails for older vs young adults: On the interplay of weapons of influence and life domains in predicting susceptibility to phishing. In *Proc. of CHI* (2017).

[53] PATTINSON, M. R., JERRAM, C., PARSONS, K., MCCORMAC, A., AND BUTAVICIUS, M. A. Why do some people manage phishing emails better than others? *Inf. Manag. Comput. Security*, 1 (2012), 18–28.

[54] PEREZ, S. Recently confirmed myspace hack could be the largest yet. TechCrunch, 2016. `https://techcrunch.com/2016/05/31/recently-confirmed-myspace-hack-could-be-the-largest-yet/`.

[55] PERLROTH, V. G. Yahoo says 1 billion user accounts were hacked. The New York Times, 2016. `https://www.nytimes.com/2016/12/14/technology/yahoo-hack.html`.

[56] POSTEL, J. B. Simple mail transfer protocol, 1982. `https://tools.ietf.org/html/rfc821`.

[57] PRAKASH, P., KUMAR, M., KOMPELLA, R. R., AND GUPTA, M. Phishnet: Predictive blacklisting to detect phishing attacks. In *Proc. of INFOCOM* (2010).

[58] RAMACHANDRAN, A., FEAMSTER, N., AND VEMPALA, S. Filtering spam with behavioral blacklisting. In *Proc. of CCS* (2007).

[59] SCHECHTER, S. E., DHAMIJA, R., OZMENT, A., AND FISCHER, I. The emperor's new security indicators: an evaluation of website authentication and the effect of role playing on usability studies. In *Proc. of IEEE S&P* (2007).

[60] SHENG, S., HOLBROOK, M., KUMARAGURU, P., CRANOR, L. F., AND DOWNS, J. Who falls for phish?: A demographic analysis of phishing susceptibility and effectiveness of interventions. In *Proc. of CHI* (2010).

[61] SUNSHINE, J., EGELMAN, S., ALMUHIMEDI, H., ATRI, N., AND CRANOR, L. F. Crying wolf: An empirical study of ssl warning effectiveness. In *Proc. of USENIX Security* (2009).

[62] THOMAS, K., LI, F., ZAND, A., BARRETT, J., RANIERI, J., INVERNIZZI, L., MARKOV, Y., COMANESCU, O., ERANTI, V., MOSCICKI, A., MARGOLIS, D., PAXSON, V., AND BURSZTEIN, E. Data breaches, phishing, or malware? understanding the risks of stolen credentials. In *Proc. of CCS* (2017).

[63] VARGAS, J., BAHNSEN, A. C., VILLEGAS, S., AND INGEVALDSON, D. Knowing your enemies: leveraging data analysis to expose phishing patterns against a major us financial institution. In *Proc. of eCrime* (2016).

[64] VISHWANATH, A., HERATH, T., CHEN, R., WANG, J., AND RAO, H. R. Why do people get phished? testing individual differences in phishing vulnerability within an integrated, information processing model. *Decis. Support Syst. 51*, 3 (2011).

[65] WANG, J., HERATH, T., CHEN, R., VISHWANATH, A., AND RAO, H. R. Research article phishing susceptibility: An investigation into the processing of a targeted spear phishing email. *IEEE Transactions on Professional Communication 55*, 4 (2012), 345–362.

[66] WANG, J., LI, Y., AND RAO, H. R. Overconfidence in phishing email detection. *Journal of the Association for Information Systems 17*, 1 (2016).

[67] WASH, R., AND COOPER, M. M. Who provides phishing training? facts, stories, and people like me. In *Proc. of CHI'18* (2018).

[68] WHITTAKER, C., RYNER, B., AND NAZIF, M. Large-scale automatic classification of phishing pages. In *Proc. of NDSS* (2010).

[69] WU, M., MILLER, R. C., AND GARFINKEL, S. L. Do security toolbars actually prevent phishing attacks? In *Proc. of CHI* (2006).

[70] ZHANG, B., WU, M., KANG, H., GO, E., AND SUNDAR, S. S. Effects of security warnings and instant gratification cues on attitudes toward mobile websites. In *Proc. of CHI* (2014).

[71] ZHANG, Y., EGELMAN, S., CRANOR, L., AND HONG, J. Phinding Phish: Evaluating Anti-Phishing Tools. In *Proc. of NDSS* (2007).

[72] ZHANG, Y., HONG, J. I., AND CRANOR, L. F. Cantina: a content-based approach to detecting phishing web sites. In *Proc. of WWW* (2007).

Appendix A – Spoofing Target Domains

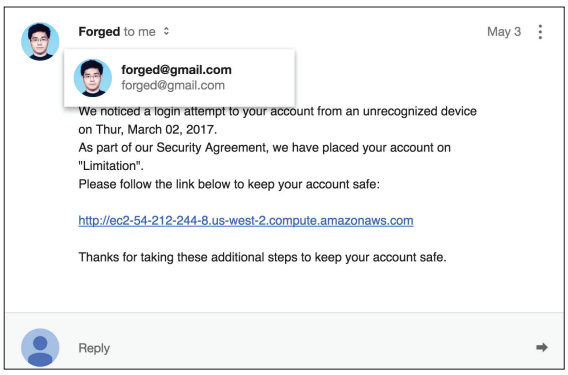
Table 7 lists the 30 domains used by the end-to-end spoofing experiment as the spoofed sender address. The domains per category are selected randomly from Alexa top 5000 domains.

None: No SPF/DKIM/DMARC (10)
thepiratebay.org, torrent-baza.net, frdic.com, chinafloor.cn, onlinesbi.com,4dsply.com, peliculasflv.tv, sh.st, contw.com anyanime.com
Relaxed: SPF/DKIM;DMARC=none (10)
tumblr.com, wikipedia.org, ebay.com, microsoftonline.com, msn.com, apple.com, vt.edu, github.com, qq.com, live.com
Strict: SPF/DKIM;DMARC=reject (10)
google.com, youtube.com, yahoo.com, vk.com, reddit.com, facebook.com, twitter.com, instagram.com, linkedin.com, blogspot.com

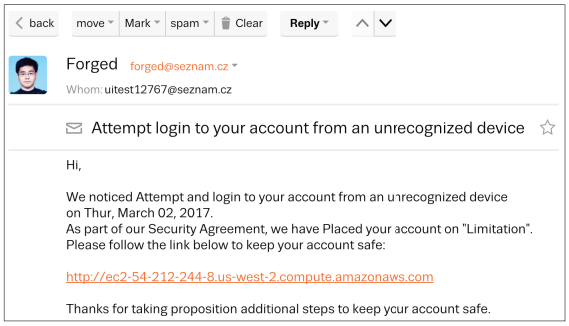
Table 7: Spoofed Sender Domain List.

Appendix B – Other Vulnerabilities

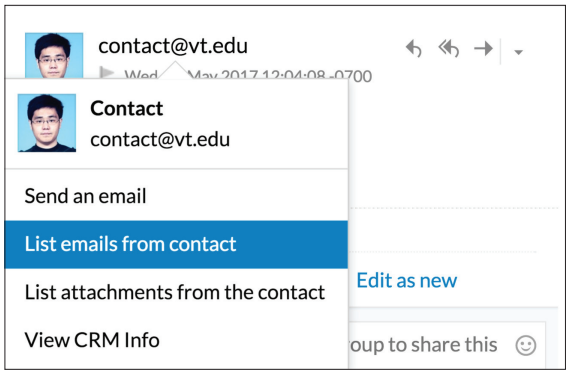
We find that 2 email services "`sapo.pt`" and "`runbox.com`" are not carefully configured, allowing



(a) Google Inbox profile photo (same-domain spoofing)



(b) Seznam profile photo (same-domain spoofing)



(c) Zoho profile photo and email history (spoofing a contact)

Figure 11: Examples of misleading UIs (profile photo, email history, namecard).

an attacker to piggyback on their mail servers to send forge emails. This threat model is very different from our experiments above, and we briefly describe it using Figure 1. Here, the attacker is the sender MUA, and the vulnerable server (*e.g.*, `runbox.com`) is the sender service. Typically, Runbox should only allow its users to send an email with the sender address as "`{someone}@runbox.com`". However, the Runbox's server allows a user (the attacker) to set the "`MAIL FROM`" freely (without requiring a verification) in step ❶ to send forged emails. This attack does not help the forged email to bypass the SPF/DKIM check. However, it gives the attacker a *static and reputable* IP address. If the attacker aggressively sends malicious emails through the vulnerable mail server, it can damage the reputation of the IP. We have reported the vulnerability to the service admins.

Appendix C – Misleading User Interface

Figure 11 shows three examples of misleading UI elements. Figure 11(a) and 11(b) show that when an attacker spoofs a user from the same email provider as the receiver, the email provider will automatically load the profile *photo* of the spoofed sender from its internal database. In both Google Inbox and Seznam, the forged emails look like that they were sent by the user "Forged", and the photo icon gives the forged email a more authentic look. Figure 11(c) demonstrates the misleading UIs when the attacker spoofs an existing contact of the receiver. Again, despite the sender address (`contact@vt.edu`) is spoofed, Zoho still loads the contact's photo from its internal database. In addition, users can check the recent email conversations with this contact by clicking on the highlighted link. These elements make the forged email look authentic.