Email Address Mutation for Proactive Deterrence Against Lateral Spear-phishing Attacks

Md Mazharul Islam[1], Ehab Al-Shaer[2], and Muhammad Abdul Basit Ur Rahim[1]

[1] University of North Carolina at Charlotte, NC 28223, USA
{mislam7,mabdulb1}@uncc.edu
[2] INI/CyLab, Carnegie Mellon University, Pittsburgh, PA 15213, USA
ehab@cmu.edu

Abstract. Email spear-phishing attack is one of the most devastating cyber threat against individual and business victims. Using spear-phishing emails, adversaries can manage to impersonate authoritative identities in order to incite victims to perform actions that help adversaries to gain financial and/hacking goals. Many of these targeted spear-phishing can be undetectable based on analyzing emails because, for example, they can be sent from compromised benign accounts (called lateral spear-phishing attack).
In this paper, we developed a novel proactive defense technique using sender email address mutation to protect a group of related users against lateral spear-phishing. In our approach, we frequently change the sender email address randomly that can only be verified by trusted peers, without imposing any overhead or restriction on email communication with external users. Our Email mutation technique is transparent, secure, and effective because it allows users to use their email as usual, while they are fully protected from such stealthy spear-phishing.
We present the Email mutation technique (algorithm and protocol) and develop a formal model to verify its correctness. The processing overhead due to mutation is a few milliseconds, which is negligible with the prospective of end-to-end email transmission delay. We also describe a real-world implementation of the Email mutation technique that works with any email service providers such as Gmail, Apple iCloud, Yahoo Mail, and seamlessly integrates with standard email clients such as Gmail web clients (mail.google.com), Microsoft Outlook, and Thunderbird.

Keywords: Lateral Spear-phishing Attack · Spoofing Attack · Email Phishing · Targeted Attack · Moving Target Defense.

1 Introduction

In recent years, email spear-phishing becomes the most effective cyber threat against individual and business victims. It has been reported that 90% of data breaches in 2017-2018 included a phishing element, 74% of public sector cyber-espionage, and 64% of organizations' attacks involve spear-phishing attacks,

where 30% of these attacks are targeted [4]. Over $26B has been lost to spear-phishing and account takeover in 2019 [1]. Only in the US, 71.4% of phishing attacks and data breaches associated with nation-state or state-affiliated actors used spear-phishing [29].

Unlike phishing, where adversaries send generic emails with malicious attachments or links to a massive number of users indiscriminately hoping that someone will take the bait, spear-phishing is more targeted. In spear-phishing attacks, adversaries impersonate key personnel or authoritative identities in order to incite victims to perform such actions that help them to gain certain goals. Adversaries carefully select their targets and send them well-crafted phishing emails that closely resemble what they usually receive. Mostly, these attack emails mimic a familiar style and signature of a known person to the victims, where both the email headers and body merely deviate from benign emails. Yet, the email contains a 'lure' that is convincing enough to engage the victim into an 'exploit' [13].

A common technique for spear-phishing attack is to spoof the name or email address of the sender, known as *source spoofing*, to convince the victim that the email is sent from a trusted source [13]. Solutions like SPF [19], DKIM [8], and DMARC [21] that verifies the sender authenticity may prevent email source spoofing [14]. However, adversaries are continuously evolving and adapting their attack strategies. As a result, a more stealthy variation of spear-phishing attack has been encountered recently, known as *lateral spear-phishing* attack, where adversary uses compromised email accounts of someone either socially or professionally connected with the victim to initiate the phishing email [12]. These phishing emails are very hard to detect because of the cleverly crafted content created by deep analyzing the prior conversation with the victim from that compromised account. Therefore, adversaries inherently win the cyber game against defenders in the lateral spear-phishing attack by evading any existing security regarding sender email authentication as the phishing email coming directly from a legitimate account and defeating behavioral anomaly detectors by accessing human anchoring as the email seemingly composed by a trusted entity [12]. These facts motivate our research to develop a proactive mechanism for protecting the number one targeted attack vector, email.

The current state of the art for detecting lateral spear-phishing emails mainly depends on email headers and body [5,10–13,15,18,27]. These defense techniques require users' historical data to model a behavioral profile for the sender or receiver in order to detect the anomalous behavior of the phishing email. They also depend on analyzing the content of phishing emails searching for malicious URLs or attachments, domains with low reputation, etc. However, spear-phishers can easily evade detection by mimicking users' behavior (from previous emails) and avoiding the use of bad features [12].

To address these limitations, we developed a novel moving target defense technique called sender Email address Mutation (EM) to proactively protect a group of users against lateral spear-phishing and spoofing attacks. EM is developed as a cloud-based service that can be easily integrated with existing email

infrastructure for any organization to offer scalable email protection against spear-phishing with minimal management overhead. It deploys a secure gateway in the cloud that works transparently between end-users and email service providers. EM defends a group of socially or professionally connected members, called the VIP users. It creates a number of random *shadow* email addresses (accounts) associated with each VIP user besides their actual email address. These shadow email addresses are used as the sender for email delivery but are hidden to both end-users.

While two VIP members communicate with each other through EM, the email first goes to the secure email mutation gateway (EMG) in the cloud, where EMG translates the sender email address to a shadow email address corresponding to the sender before forwarding it. Similarly, when the receiver VIP user fetches that email, the EMG verifies the shadow email address and delivers the email to the recipient, if the verification is successful. Therefore, knowing the public email address will not be sufficient to attack the VIP users. Spear-phisher adversaries must correctly guess the current shadow email being used by each individual user in order to successfully impersonate a VIP user in an email sent to another VIP user. While EM achieves this protection between VIP users, it also maintains the email open communication model by allowing VIP users to receive and send emails to any external users without any restriction. Thus, EM can protect a group of socially or organizationally connected (VIP) users from any phishing emails that impersonate a VIP user even if the email is coming from a compromised VIP email account, without impacting users' usability or interaction with external users.

PGP [6], S/MIME [24], Two-factor authentication (2FA) [3] can be used to authenticate email senders and prevent email account hijacking. However, these techniques are not widely used in practice due to many users' transparency, usability, and management challenges [25, 26, 28]. For instance, PGP signature and encryption obfuscate the plaintext emails into cyphertext, immediately losing the content's visibility. Therefore, existing IDS and content-based behavioral analysis tools can not work with PGP encryption. Furthermore, PGP requires user training on Public key infrastructure (PKI) and maintains complex key management systems. Moreover, PGP signatures in email can be spoofed [23]. EM provides an alternative proactive mechanism for the majority of email users who are not using PGP and/or 2FA to protect against spear-phishing without compromising transparency, usability, or manageability. Therefore, although EM does not use a cryptographic approach like PGP or 2FA, it can provide comparable protection while maintaining high usability and deployability.

Our key contribution is three-fold:

- First, we introduced a novel protocol called EM, as a proactive defense against highly stealthy lateral spear-phishing attacks.
- Second, we verified that the EM protocol is valid and can be integrated with any existing email service provider.
- Third, we implemented the EM system (code available on GitHub) and deployed it in a real-world environment without imposing any usability or performance overhead on users or service providers.

2 Related Work

A vast amount of research has been done to detect phishing and spear-phishing attacks [9–15, 18, 27]. The majority of these works depend on email content or headers. For instance, Ho et al. presented a learning-based classifier to detect lateral spear-phishing by seeking malicious URLs embedded in the phishing email [12,13]. However, these solutions will not work against motivated adversaries trying to evade detection simply just by not adding any malicious ULR or no URL at all in the phishing email.

Spear-phishing detectors like EmailProfiler [10] and IdentityMailer [27] also depend on email headers and body to build behavioral profiles for each sender based on stylometric features in the content, senders writing habits, common recipients, usual email forwarding time, etc. New emails get compared with the profile to measure the deviation determining whether they are spear-phishing email or not. These solutions can not detect lateral spear-phishing emails when the contents are carefully crafted to avoid deviation from the norm. Moreover, they show a high false-positive rate (10%), which becomes unacceptable when it comes to the large volume of emails in a real-world enterprise network.

Gascon et al. [11] proposed a context agnostic spear-phishing email detector by building behavioral profiles for all senders in a given user mailbox. They create such profiles from the common traits a sender follows while composing an email, like attachments types, different header fields, etc. However, in the lateral spear-phishing attack, email headers do not deviate from the usual structure as it is coming from a legitimate account. In addition, building profiles for each sender can induce massive overhead in large scale deployment.

Existing sender authentication protocols such as SPF [19], DKIM [8], and DMARC [21] can not detect lateral spear-phishing emails because they are not spoofed and composed from valid email accounts. Other solutions, such as signing emails with PGP [6], S/MIME [24], or 2FA [3] can prevent the lateral spear-phishing attack. Unfortunately, these techniques are not widely used because of usability, manageability, and transparency issues [25, 26, 28]. Moreover, a recent study showed that PGP signatures in the email could be spoofed as well [23].

3 Threat Model

3.1 Attack Taxonomy.

In the lateral spear-phishing attack, adversaries send phishing emails to victims from a compromised account. To make such attacks trustworthy and effective, adversaries carefully choose those compromised accounts that are closely related to the victims, such as employees from the same organization [12]. Therefore, the attacker easily bypasses traditional email security systems like sender authentication, as the email is come from a legitimate account and make the victim fall for the attack, as it is seemingly composed by a person they already trust.

```
From: Alice <alice@org.com>
To: Bob <bob@org.com>
Subject: February, 2020 Meeting Budget (Event venue booking)
Hi Bob,
Process wire transfer of $100,543 to Trudy (account no. 5648132796, routing no. 026001234) to
finalize upcoming  event venue bookings. Send me an invoice of that transaction ASAP, thanks.
Alice
CEO, org.com
```

Listing 1: A carefully crafted lateral spear-phishing email sends to Bob from a compromised account Alice, without any malicious attachments or URLs.

Listing 1 depicts an example of lateral spear-phishing email. Adversary Trudy compromises the email account of Alice, CEO of an organization (`org.com`). By examining her inbox, Trudy obtains that Alice directed Bob, finance department head of `org.com`, to make some wire transactions for arranging an upcoming business meeting. Exploiting this analysis, Trudy composes a phishing email from Alice's email account to Bob, directing him to make a wire transaction in Trudy's bank account. These types of lateral phishing are crafted carefully by observing previous emails and may not contain any malicious attachments or URLs that make it very hard to detect.

3.2 Attack Model

The EM protocol detects lateral spear-phishing and spoofing attacks where adversaries send phishing emails to the victim from a compromised benign email account or impersonate a benign person that the victim already trusts. Compromising an email account means adversary gain access only to that email account, not the physical machine such as laptop, desktop, or cell phone itself of the user. Moreover, any compromised account who never communicated with the victim before can hardly be successful in exploiting the victim. Therefore, EM solely focuses on compromised accounts and impersonated entities that are connected with the victim, e.g., employees from the same organization or different organization, but communicates frequently. The people who use EM to protect themselves against spear-phishing attacks are denoted as VIP users. To launch a lateral spear-phishing attack, an adversary needs to send the phishing email from a compromised account, Alice, for instance, whom Bob already connected with. EM can protect Bob against such an attack if both Bob and Alice are agreed prior to use EM; therefore, they are in the VIP user list. EM also protects VIP users from spoofing if the adversary impersonates any of the VIP users in the phishing email.

4 Email Mutation System

4.1 Overview

Figure 1 depicts an overview of sender email address mutation, where two VIP users Alice and Bob from an organization (`org.com`), agreed to use EM to

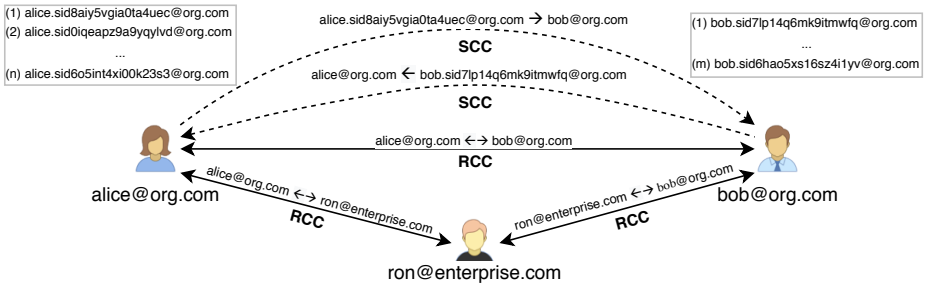


Fig. 1: Email Mutation overview. Alice and Bob send emails using their shadow email addresses (dashed line, single arrow).

protect themselves against lateral spear-phishing attacks. Previously, they communicate with each other using their real email address (double arrow solid line), called Real channel communication (RCC). However, after EM starts, each VIP user gets a list of shadow email accounts. For instance, Alice and Bob get n and m number of shadow email accounts (addresses), respectively. Thus, each new email Alice composes for Bob now uses a different shadow email address as a sender instead of her real email address, and the modified (mutated) email is forwarded to Bob (dashed line single arrow). This is called Shadow channel communication (SCC). Sending an email in SCC by mutating the sender email address is known as *mutation*. When Bob receives such an email, the shadow email address gets verified for lateral spear-phishing detection. This is called *verification*. Similarly, when Bob composes a new email for Alice, the email is forwarded through SCC by mutating Bob's real email address to one of his shadow email addresses. Although Alice and Bob use shadow email addresses as the sender to communicate with each other through SCC, they use their real email address to communicate with external users (non-VIP), e.g., Ron from `enterprise.com`. VIP users group can comprise people from different organizations having different domains. For instance, John (not shown in the figure) from `gov.com` can be a VIP member with Alice and Bob from `org.com`.

Shadow Email Accounts. The shadow emails are a list of pre-created email accounts assigned to all VIP users but being kept hidden from them. These accounts are only used in email transmission as a sender address. Only EMG conducts with these email accounts. Depending on the impacts, the number of shadow email addresses assigned to a VIP user varies. EM is flexible to the creation of shadow email accounts as long as the shadow email domain is the same as the real email domain. However, in our experiment, we used a prefix "sid" (shadow ID) in the shadow email address to make a clear difference with the real email address. A possible shadow email address may look like: *real.email.address.x@domain*, where x is at least 16 byte long random alphanumeric sequence. For instance, *alice.sid8aiy5vgia0ta4uec@org.com* can be one of the shadow email addresses for Alice's real email address *alice@org.com*, where x = *sid8aiy5vgia0ta4uec*.

4.2 Architecture

The *mutation* and *verification* happens in the cloud by mutation gateways (EMG). Figure 2 illustrates the architecture of EM. Clients can use multiple devices such as laptops, desktop, or cell phones for accessing emails; therefore, EM provides an EM agent (EMA) for each of the devices.

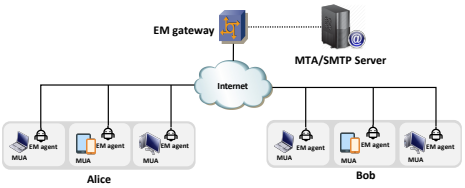


Fig. 2: Email mutation architecture.

While sending an email, the agent delivers the email to the EMG for mutation. After mutation, the EMG forwards the mutated email to corresponding mail servers (SMTP/MTA). Similarly, while fetching a new email, the agent receives it from the EMG. The EMG first gets the email from the mail server and then verifies it to detect a lateral spear-phishing attack before responding to the agent. In large enterprise networks, multiple EMGs can be used for load balancing.

4.3 Algorithm

The VIP users supposedly send emails to each other, which use as ground truth G next time they send any new emails. For instance, when a VIP user i sends an email to another VIP user j, the last l emails between them will be used as ground truth $G_{i,j}$ to generate a mutation ID, *mID*. By indexing the *mID*, a shadow email address gets selected from a secret arrangement of shadow email addresses S_i assigned for the sender i. The shadow email address then used to forward the email. Similarly, as the receiver j has the identical ground truth, j can generate the exact *mID* to find the same shadow email address from S_i for verification.

Algorithm 1 shows the pseudocode of shadow email address selection. A hash function SHA-512 is used to get the digest of $G_{i,j}$, which then modulo with the size of S_i to select the current shadow email index, *mID*. Although from a compromised VIP user account, the adversary can achieve the ground truth $G_{i,j}$ to calculate *mID*, yet can not get the correct shadow email address because of not having the secret arrangement of S_i. Therefore, the adversary can not send an email with the right shadow email address, which immediately gets detected by the EMG.

Algorithm 1 Shadow Selection

```
1: procedure SELECTSHADOW(G_{i,j}, S_i)
2:     h ← SHA-512(G_{i,j})
3:     mID ← h mod len(S_i)
4:     shadow ← S_i[mID]
5:     return shadow
```

4.4 Protocol

Communication between VIP users. Figure 3 explains the EM protocol through email communication between VIP users Alice and Bob. (1) Alice composes an email to Bob {*from: alice@org.com, to: bob@org.com*}. (2) Alice's EMA

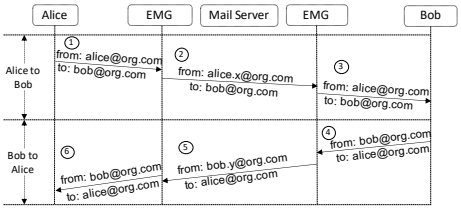


Fig. 3: EM protocol, demonstrating email communication between VIP users.

delivers the email to EMG, where EMG uses the ground truth between them in algorithm 1, to select a shadow email address for Alice. Assume that the selected address is *alice.x@org.com*. Therefore, the EMG forwards the email to the mail server as {*from: alice.x@org.com, to: bob@org.com*}. (3) When Bob's EMA fetches for a new email, EMG receives the email from the mail server and deduce that the sender address is one of Alice's shadow email addresses. Therefore, EMG uses the ground truth between Bob and Alice in algorithm 1 to select the current shadow email address for Alice. If the retrieved address matches *alice.x@org.com*, the email is benign. Otherwise, it is phishing. EMG delivers the benign email to Bob's EMA as {*from: alice@org.com, to: bob@org.com*}. Bob receives the email as how Alice composed it, making the whole EM mechanisms transparent to end-users. The replies from Bob to Alice is similar to the above three steps. The only secret in the protocol is the arrangement of the sender shadow emails.

Communication with External Users. EM only protects VIP users. Therefore, the EMG bypasses the following emails with external (non-VIP) users to decrease the overall email traffic processing:

No Mutation. A VIP user sends an email to a non-VIP user. Formally:

$$\{sender : x,\ recipient : y;\ where,\ x \in R \text{ and } y \notin R\}$$

where R is the list of real email addresses of all VIP users.

No Verification. A VIP user receives an email from a non-VIP user. Formally:

$$\{sender : x,\ recipient : y;\ where,\ x \notin R \text{ and } y \in R\}$$

4.5 Identifying Lateral Spear-phishing Attack

In EM, the legitimate email communication between VIP users happens through SCC, where the sender address is always a valid shadow email address. Therefore, EMG detects an incoming email as phishing while fetching new emails that have a real email address of a VIP member as the sender's address. However, the adversary may send phishing emails by guessing or randomly generating a shadow email address to bypass EM. We call such phishing attempts as EM engineering attack. To formalize the detection process, let's assume that R is the real email address list, and $\overline{S}$ is the set of shadow email address lists of all VIP users.

Lateral Spear-phishing and Spoofing Attack. By compromising a VIP user's email account or impersonating a VIP user, the adversary sends a phishing email to another VIP user. Formally:

$$\{sender : x,\ recipient : y;\ where,\ x, y \in R\}$$

That means both the sender and receiver email address is enlisted in the VIP user list. EMG immediately detects such an email as a phishing email.

EM Engineering Attack. The adversary sends a phishing email to any VIP user by randomly *guessing* a shadow email address as the sender's address. Formally:

$$\{sender : x,\ recipient : y;\ where,\ x \in \overline{S}\ and\ y \in R\}$$

To evade EM, adversaries may randomly guess or mimic the mutation mechanism to generate a legitimate shadow email address. However, EM creates shadow email addresses from a space of at least 16 byte long alphanumeric sequence. Therefore, the probability of guessing a correct shadow email address is $1/2^{128}$, which is nearly zero.

5 Email Mutation – Challenges and Solutions

Handling Multiple Shadow Email Accounts. In EM, each VIP user has a set of shadow email accounts for sending emails to another VIP user. However, VIP users only discern about their real email account. Therefore, sending emails from multiple shadow accounts and keeping track of all sent emails into one real email account is challenging. EM overcome this challenge by following means. First, shadow email accounts only used for sending emails while the receiver email address will always be the real email address. Therefore, each VIP user receives all emails into their real email account inbox. Second, while forwarding an email from a shadow account, the EMG uses an IMAP `APPEND` command to populate that email into the real email sent-box after a successful email delivery to the recipient. Thus, the real account gets a trace of email delivery. If an email gets bounced, the EMG sends the bounce email back to the real email account.

Improving Email Mutation Usability. Existing phishing detectors similar to EM mostly suffer because of low usability. For instance, PGP requires user training on public-key cryptography and the PGP tool itself [25, 26]. PGP encryption removes the visibility of the email from end-users. Whereas, EM does not distort the generic user experience of using emails. Every operation such as mutation, verification, and shadow email address communication entirely segregated from the end-users and processed by the cloud EMGs. Users only need to use EMAs, for instance, sending an email by pressing the new "Send email with mutation" button beside the regular "Send" button in the Gmail web client (`mail.google.com`) illustrated in figure 4a. EM does not modify or add anything in the email body or headers makes it transparent to mail servers as well. Therefore, EM can be used with any email service provider and email clients without further usability and configuration overhead. The transparency of EM also makes it compatible to work combining with other email security solutions (even with PGP [6] or S/MIME [24]) and cyber agility frameworks [16, 17].

Preserving User Privacy. The secure cloud-based gateways in the EM does not violate the end-to-end user email privacy because of the following reasons. Firstly, EMG does not keep any copy of the email. It just either mutates or

verifies the sender's email address if the communications happen between two VIP users. All other emails get bypassed. Secondly, EMA connects with EMGs through secure channels (SSL/TLS) to avoid unauthorized data access during transmission. Finally, the organization of the VIP members can maintain their own EMGs to preserve data privacy. The secret shadow email lists can not be retrievable from any EMGs. Therefore, in a cross-enterprise EM system, EMG from one organization can not reveal the shadow email list of another organization. In recent days cloud-based secure email gateways are becoming popular because of their swift, robust, and effective attack detection with minimal management overhead. Therefore, many organizations are adopting such solutions from Cisco, Microsoft, Barracuda, Mimecast, and others [2].

Adding Custom Email Fields is Insufficient. Adding just a new field in the email headers (such as X-headers [7]) or custom trace on the email body for sender authentication will not help to detect the lateral spear-phishing attack. Because adding any extra information into an email will immediately eliminate its transparency to both the client and the mail server. Second, adversaries may corrupt such additional data by adding noises into it, which will cause an interruption in regular email communication, raising high false-positive rates, and opening new security loopholes into the detection system. Finally, motivated adversaries can craft such fields by carefully observing historical emails. To overcome these challenges, EM uses a random selection of the sender email address (shadow address) for each new email delivery without adding anything into the email. This makes the solution transparent to both end-users and mail servers, but hard for adversaries to guess a correct shadow email address.

Addressing Asynchronous Ground Truth Problem. If a VIP user deletes any email from his inbox/sentbox that was sent by another VIP user, then the ground truth between them becomes asynchronous. To solve this problem, EMG keeps the hashed (SHA-512) digest of the last l number of emails between two VIP users. Therefore, EMG stores a maximum of $(v-1)$ hashed ground truth for each VIP user, where v is the number of all VIP users, to prevent asynchronous ground truth problems. Moreover, sender and receiver EMGs perform this hash storing operation asynchronously while sending and/or receiving an email. Thus, the *synchronization* does not require any communication between EMGs.

Handling Insider Attack. A novel contribution of EM is that it can protect VIP users from insider attacks. For instance, John is a VIP user who stoles Alice's email account credentials. Then, John uses his EMA to send a phishing email to Bob impersonating Alice. Formally, an attacker i compromises an email account j and uses i's EMA to send emails to k impersonating j, where $i, j, k \in L$ and L is the list of all VIP users. EM solves this problem by following: every VIP user's EMA is synchronized with its corresponding EMG instance through a unique authentication token (see section 6.1 for details). That means John's EMA can get only his instance of EMG; therefore, it will work only for John himself, not for Alice or Bob. The EMG keeps track who is forwarding the email by examining the authentication token of the EMA, and then verifies if that EMA is associated with the user or not. If the EMA is not assigned for that

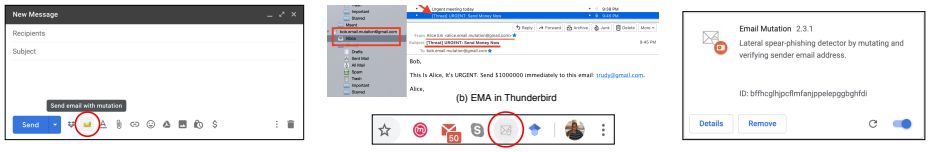


Fig. 4: EMA in Google Chrome Browser and Thunderbird, (a) "Send email with mutation" button in `mail.google.com`, (b) EMA in Thunderbird, (c) EMA icon at Chrome Menu Bar, and (d) EMA in Chrome Extension Panel.

particular user but delivers an email anyhow, then EMG identifies that email as an insider attack.

Minimizing Shadow Email Account Overhead. The shadow email accounts are only meant to send emails as sender. These accounts do not receive any emails. Creating multiple shadow accounts for VIP users does not increase any inbound email traffic. Besides, these accounts do not impose any memory overhead. Therefore, they create negligible overhead on the service provider. The shadow email addresses can be selected from a 16-byte long (at least) sequence. This ensures no collision between shadow email with real email accounts. Additionally, unique keywords such as "sid" (section 4.1) can be used in shadow accounts creation to make a fine difference from real email accounts

6 Scalable Implementation and Security Measurement

Evaluating EM in a large scale network requires a scalable implementation that is compatible with any existing email clients and email service providers. Moreover, the security measures of each component are necessary to reduce the risk factor in terms of user privacy and data breach. We measure these matrices for the primary components of EM: EMA and EMG.

6.1 Email Mutation Agent

Security Measures. EMA operates alongside with regular Mail user agent [20] or email clients only to deliver or receive new emails from EMGs. It does not communicate with the mail or SMTP server. Therefore, EMAs neither have any storage to collect emails nor requires the user email account credential. The communication between EMA and EMGs happens through a secure channel (SSL/TLS) to protect data breaches during the transaction.

Implementation. Usually, clients use multiple devices, such as cell phones, laptops, desktop, and more, to access their email accounts. Therefore, EMA needs platform-oriented implementation to work on different devices as well. We implemented three types of EMA for three different platforms, 1) browsers extension for web clients that will work in laptops, desktop, where a web browser can run. 2) Shell and python scripts to configure email clients such as Outlook, Thunderbird, and 3) email client (android/iOS) app for cell phones and tabs.

Figure 4 shows different implementation of EMAs, such as browser extension (4a) and Thunderbird client (4b). The Chrome browser extension adds a new button called "Send email with Mutation" (figure 4a) alongside with the regular "Send" button that `mail.google.com` provides.

Distribution of EMA. A VIP user can get an EMA from their system admin or download it from the web. The admin gives a unique authentication token to each VIP user for their first use of EMA to subscribe with the EMGs. Later on, using that token, they can connect with their corresponding EMG from different EMAs. This ensures users' flexibility to use EMA from different devices and different locations (e.g., public networks). Users can reset the token anytime.

6.2 Email Mutation Gateway

Security Measures. EMG inspects email for detection and modifies for mutation. It does not maintain any mailboxes for users. When a VIP user subscribes with EMG, it creates an instance for that user. So that later on, the same user can connect with the EMG from EMAs in different devices. EMG keeps the arrangement of the shadow email lists secret. Therefore, from an EMA, VIP users can not retrieve their shadow email list. After a certain mutation interval t seconds, EMG rearranges all the secret shadow email lits of VIP users randomly. Besides, an instance of EMG given to a VIP user is not shareable by other VIP users through EMAs, meaning Alice can not use the EMG instance of Bob from her EMA. This protects the insider attacks because using her EMA and EMG instance, Alice can not send emails as Bob, considering that Alice compromises Bob's email account. Cloud-based solutions like EMGs are secured, and many providers like Amazon, Microsoft, Cisco, Barracuda, Mimecast, and more are nowadays providing secure cloud email gateways for phishing detection [2].

Implementation. We implemented the EMGs as an inbound and outbound Mail transfer agent [20] that works as an SMTP relay to mutate outgoing emails and proxy gateway to check incoming emails for spear-phishing detection. We use python libraries such as `smtpd` and `imaplib`, to implement the relay server and Django framework to make EMG a web service. The code is available on GitHub.

7 Email Mutation Verification and Evaluation

7.1 System Verification

EM is a new technique; therefore, it is necessary to ensure the design correctness before implementation and deployment over real network. Model checkers help to formally specify, model, and verify a system in the early design phase to find any unknown behavior if it exists. This section presents the modeling of individual components, their interaction, and verification of EM using model checking tool UPPAAL [22]. The comprehensive system is modeled using timed

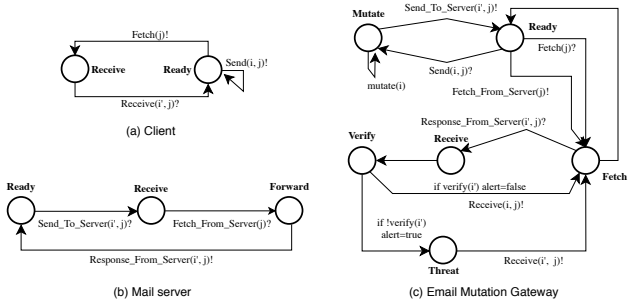


Fig. 5: State machine diagrams of different components in EM.

automata and verified against user-defined temporal properties. We have illustrated the components of EM using the state machine diagram of the system modeling language, where the circle represents the state, and the arrow shows the transition among the states. The transitions are annotated with channels, guards, and actions. Interaction between components using channel $(!, ?)$ where, "!" and "?" means the sending and receiving signals, respectively.

Modeling of Client and Mail Server. Figure 5a illustrates the functionality of a client. The client uses the sending signal $Send(i, j)!$ to forward a newly composed email to the EMG, where i is the sender address, and j is the receiver address. Using $Fetch\_Email(j)!$, client j request to fetch any new email from EMG. As a response, the client receives a new email from EMG by the receiving signal $Receive(i, j)?$, where i is the sender address, and j is the receiver address. Figure 5b shows the basic functionality of a mail server. The channels $Send\_To\_Server(i, j)?$, $Fetch\_From\_Server(j)?$, and $Response\_From\_Server$ $(i, j)!$ represent the transitions for receiving a new email, receiving fetching request for new emails, and responding new emails respectively.

Modeling of Gateway. Figure 5c describes the functionality of the mutation gateway (EMG). EMG receives a new email from clients through the receiving signal $send(i, j)?$ and mutates the sender address i to i' using the function $mutate(i)$. Then forwards the email to the mail server using signal $Send\_To\_Serv$ $er(i', j)!$. EMG goes the *Fetch* state after receiving a $Fetch(j)$ signal form client j and seeks new emails from the mail server by the signal $Fetch\_From\_Server(j)!$. As a response, EMG receives new emails by the receiving signal $Response\_From$ $\_Server(i', j)?$, where i' is the (mutated) sender address, and j is the receiver address. After that, EMG verifies i' by the function $verify(i')$. If verification pass, then the email will be delivered to the client through the $Receive(i, j)?$, where i is the real address of i'. Otherwise, the email gets flagged as a threat. In case of suspicious email, the invariant *alert* is set to true; otherwise, it is set to false. Here, the state *Threat* is presented to flag the email.

Modeling of Adversary. Using channel $Send\_To\_Server(l, j)?$ adversaries send email to recipient j, where l is the adversary chosen sender address. If adversaries make a successful guess, their email will be delivered to the client.

Property	CTL	Result
Reachability	$A[] := \forall\ i \in all\_emg$ $(emg_i.verify\ \wedge\ !emg_i.alert) \rightarrow (emg_i.ready)$	*satisfied*
Liveness	$A[] := \forall\ i \in all\_emg\ (emg_i.alert\ \rightarrow emg_i.threat)$	*satisfied*
Deadlock-freeness	$A[] := \forall\ i \in all\_emg\ (!emg_i.deadlock)$	*satisfied*

Table 1: Temporal properties to verify the correctness of Email Mutation system.

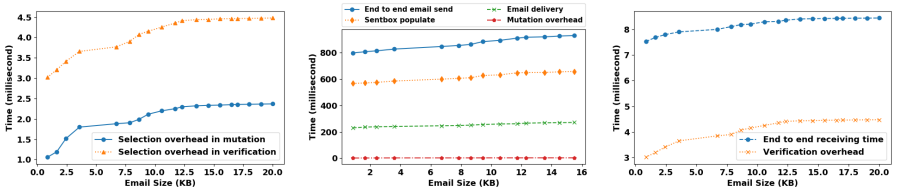
Property Verification. UPPAAL takes the model, and user-defined temporal properties as input to verify the model and generates output, either satisfied or not satisfied. UPPAAL defines temporal properties in the form of computational tree logic (CTL), which is a branch of symbolic logic. To verify EM, we defined the liveness, reachability, and deadlock-freeness as temporal properties. Table 1 describes the properties along with its UPPAAL supported CTL and the results of these properties. *Reachability* describes that every good state is reachable, and every bad state is unreachable. In EM, every benign email should be delivered to the destination. *Liveness* describes the system is progressing to achieve a specific goal. For instance, every suspicious email should be flagged as a threat. *Deadlock-freeness* ensures that the system is not stopped, and it is always progressing. The system is in deadlock sate when it stops in a state and does not proceed to other states. The CTL operator $A[]$ represents that every single state on all paths should satisfy the properties, all benign emails are delivered to the destination and every suspicious email is flagged. The reachability property ensures that no suspicious email will be delivered without a threat flag. In reachability and liveness properties, $(\forall\ i \in all\_emg)$ is used for iteration and verification of property against every single email mutation gateway (emg_i).

7.2 Performance Evaluation

We measure the performance of EM in terms of overhead added into existing email infrastructure. EM has a 100% lateral spear-phishing and spoofing detection rate. We compare EM with similar existing solutions to measure the necessity and effectiveness of the system.

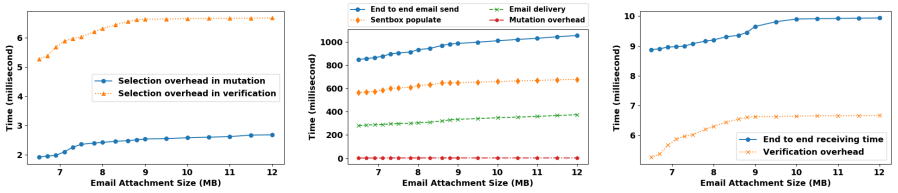
Experiment Setup. We evaluated EM in large scale enterprise networks for more than six months and protected 5,000 VIP members over five different organizations. Among them, 46 VIP member was voluntarily from Jet Propulsion Laboratory, NASA (JPL). The JPL red team sends more than half a million attack emails. The VIP members use different mail service providers, including Gmail, Microsoft Exchange, Apple iCloud, and email clients like Outlook, Thunderbird, `mail.google.com` and so on. The shadow email addresses for each VIP user was between ten to one hundred. The mutation interval t was set with different values between 60 seconds to 2 hours to rearrange the secret shadow email lists randomly. All evaluation values have been achieved from real-time email communications.

Shadow Email Selection Overhead. EMG computes shadow email addresses for mutation and verification using algorithm 1. We measure the selection over-



(a) Shadow computation. **(b)** Email mutation time. **(c)** Email verification time.

Fig. 6: EM gateway performance for mutation-verification of individual emails without attachments.



(a) Shadow computation. **(b)** Email mutation time. **(c)** Email verification time.

Fig. 7: EM gateway performance for mutation-verification of individual emails with attachments.

head of a single email over different email sizes. Figure 6a shows the selection overhead for mutation is 2.5 milliseconds, and verification is 5 milliseconds for email size range 10-20KB without attachments. Figure 7a shows the overhead is 3 milliseconds, and 7 milliseconds for email size range 7-12MB with attachments.

Mutation Overhead. While forwarding an email, EMG 1) mutates the email, 2) delivers the mutated emails to the provider mail server, and 3) populates the real email account sentbox of the sender to keep track of successful email deliveries. Figure 6b and 7b shows the overall forwarding delays over email sizes. Emails in 10-16KB sizes need 3 milliseconds for mutation, 250 milliseconds for delivery, 650 milliseconds for sentbox population, and overall 950 milliseconds to forward the email. For email sizes 7-12MB, mutation delay is 4.5 milliseconds, and overall sending time is 1.5 seconds. In both cases, the mutation overhead is 0.5% compared to the end-to-end email forwarding delay.

Verification Overhead. Figure 6c and 7c shows the end-to-end email receiving time with verification over different email sizes. Emails in 10-20KB sizes without attachments have overall 8.5 milliseconds receiving delay where the verification delay is 4.5 milliseconds, and emails in 7-12MB sizes have overall 10 milliseconds receiving delay where the verification delay is 7 milliseconds.

Email Processing Rate by an EMG. We measured the overall performance of an EMG by sending more than thousands of emails at a time to a single EMG to process mutation and verification simultaneously. Figure 8 and 9 shows the average processing delay of EM for sending an email with mutation is 1.1 seconds and receiving an email after phishing detection is 10.9 milliseconds, respectively while dealing with 5000 emails per second.

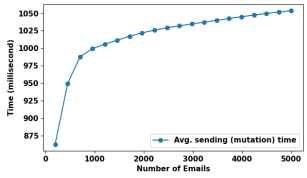


Fig. 8: Multiple email processing overhead for mutation.

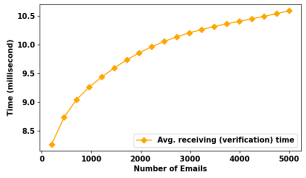


Fig. 9: Multiple email processing overhead for verification.

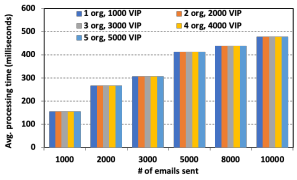


Fig. 10: EMG overhead in cross-enterprise architecture. The increasing number of organization or VIP members do not impact the overall processing time. The number of emails dealt at a time determines the overall delay.

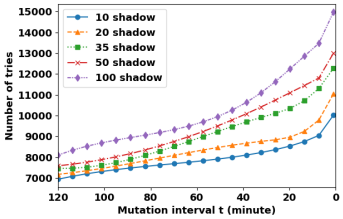


Fig. 11: EM engineering attack, the minimum number of tries adversary needs to break the EM for sending their first successful phishing email.

Cross-Enterprise Architecture Overhead. Adding different organizations into the EM system does not increase additional overheads because the operational cost of each EMG only depends on the total number of emails get processed at a time, not on the size of the VIP user list. Figure 10 depicts that the overall email processing time is the same for one organization having 1,000 VIP users to five organization having 5,000 VIP users. The delays increased when the total number of emails dealt at a time increases. Multiple EMGs can be used to balance these increasing delays.

EM Engineering Attack. If adversaries send a phishing email directly from compromised VIP user accounts or by impersonating a VIP user, they have 0% chance to evade EM. However, adversaries may try to phish by randomly guessing a shadow email address, known as EM engineering attack. A shadow email address is a 16-byte long random alphanumeric sequence, which is practically impossible to guess. Therefore, for the sake of the evaluation, we inform all valid shadow emails to the red team before launching the attack. Figure 11 depicts the detection results. With different setups of mutation parameter, the minimum number of tries adversary needs to break EM for sending their first successful phishing email varies. For instance, it takes 7,000 tries to send the first phishing email if a VIP user has 10 shadow emails and 120 minutes mutation interval. However, the tries dramatically increase to 14,500 (probability 0.000069) if the number of shadows and mutation interval changes to 100 and 1 minutes, respectively. This indicates that based on user impact, EM can increase the level of protection swiftly.

Metric	Data
Total attack emails	516,000
Lateral spear-phishing attack	153,207
Spoofing attack	145,291
EM engineering attack	201,437
Integrity attack	16,065
EM engineering attack missed	3
Integrity attack missed	167
L. spear-phishing detection	100%
Spoofing detection	100%
EM engineering detection	99.99%
EM engineering false negative	0.0015%
Integrity attack false negative	1.04%

Table 2: Detection results of EM.

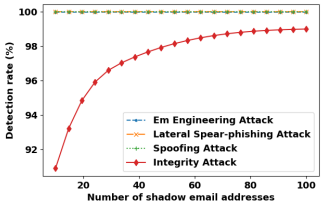


Fig. 12: The detection rate of EM over different shadow email address assigned to a VIP user. Lateral spear-phishing and spoofing detection rate is 100% for all values of shadow email addresses. The integrity attack can be detected with 99% accuracy by using 100 shadow email addresses.

Detection Results. Table 2 summarizes the performance metrics for EM. From a total of 516,000 attacks, EM detected all of the lateral spear-phishing and spoofing emails with no false positive and false negative rates. Out of 201,437 EM engineering attacks, three were successful, as because we inform prior to the attack generator red team about all valid shadow email addresses. The purpose of EM is to detect lateral spear-phishing and spoofing attacks. However, EM can detect any integrity violation in the email while two VIP user communicates. EM calculates the current shadow email address based on the hashed value of the email and prior l emails (ground truth). Therefore, any changes in the email during transmission will desynchronize the ground truth at the receiver side. Figure 12 shows the integrity attack detection rate is 99% when the shadow address for a user is one hundred. Although this attack is explicitly out of our attack model, however, we add this into the detection results to show the capabilities and completeness of EM.

Comparison with Existing System: *Learning-based.* Existing lateral spear-phishing detectors are mostly learning-oriented; they learn attack signatures, benign users' behavior from the historical data, and create a model to detect phishing emails [10–13, 27]. These solutions require myriad historical data for training, distinct attack signature (e.g., malicious URL or attachment), sufficient number of features, and often shows high false positive and false negative rates because of any lacking of these requirements. False positive means benign emails detected as phishing and false negative means a phishing email detected as benign. For instance, Ho et al. [12] trained their model over 25.5 million emails, and their attack model is limited to malicious URLs embedded in the email body. Gascon et al. [11] showed high false positive (10%) and high false negative rates (46.8%) against detecting lateral spear-phishing attacks. EM does not require any training; it is independent from email content or header and can detect lateral spear-phishing and spoofing attacks with zero false positive and false negative rates.

Tool	**Overhead** (milliseconds)			
PGP	GnuPG	Autocrypt	Enigmail	Mailvelope
	760.356	680.65	852.12	785.65
EM	Mutation	Verification		
	3	7		

Table 3: Comparison between existing popular PGP tools and EM.

Agent-based. PGP, S/MIME can ensure sender authenticity by digitally signed the email. However, these solutions are not widely used because of low usability, low transparency, and high manageability issues [25, 26]. The end-users require prior knowledge regarding public-key cryptography and proper training to use PGP tools. The encrypted cyphertext eliminates the visibility of the emails, making it incompatible to work with other security extensions such as IDS. Moreover, PGP signatures can be spoofed [23]. In contrast, EM is transparent, has a low management overhead, and highly flexible to use. The end-users do not need any prior knowledge or training to use it. Table 3 shows a comparison of EM with the existing popular PGP solutions in terms of overhead added in a generic mail transfer system. The overhead of EM is negligible (3-7 milliseconds) compared to PGP signature and encryption operations with RSA 2048 bit keys, which is vital for large scale enterprise networks where email processing rate is higher.

Authentication-protocol. Standard email spoofing detection protocols such as SPF, DKIM, and DMARC can not detect lateral spear-phishing attacks. Table 4 depicts that, all of our attack data has these security extensions. However, the lateral spear-phishing emails bypass these standard techniques as they are sent from legitimate accounts. Nonetheless, EM detects them all. Therefore, in the prevailing lacking of alternative protection against lateral spear-phishing attacks, the EM system is a valuable extension to existing defenses.

Auth. protocol	**Attack data**	**LSP detect**
SPF	100%	0%
DKIM	100%	0%
DMARC	100%	0%
EM	N/A	100%

Table 4: Existing email authentication standards failed to detect lateral spear-phishing (LSP) attacks.

8 Limitations and Future Work

One limitation of EM is that the VIP users' physical machine gets compromised or stolen, where an instance of EMA of that user is installed. In that case, EM can not protect the user. Another limitation regarding usability for EM is to get an EMA instance for every single device VIP members use to access their email account. In the future, we want to enhance EM to protect users if their device gets compromised. Moreover, we want to leverage EM on the server-side to remove the use of EMA.

9 Conclusion

In this paper, we presented a novel approach using sender email address mutation to proactively defend against the most devastating and stealthy spear-phishing

called lateral spear-phishing attacks. Our EM system guarantees the phishing emails sent from trusted users will be immediately detected. EM integrates well with existing email infrastructures, and it requires no special handling by users. EM requires an agent to be deployed on the client-side for every user and a central gateway in the cloud. The agent can be a simple plugin installed in email clients. We implemented and evaluated EM in a large scale real-world enterprise network with well-known email service providers (Gmail, for example). Our evaluation showed that EM causes 0.5% overhead on overall email transmission while detecting lateral spear-phishing and spoofing attacks. Moreover, we showed that it is very hard to break EM (probability 0.000069). Unlike the existing spear-phishing detectors, which are limited on malicious content or links, our EM can work beyond email content and headers to detect most stealthy lateral spear-phishing attacks that exploit compromised email account.

Acknowledgement

This research was supported in part by the Defense Advanced Research Projects Agency (DARPA), United States Army Research Office (ARO) and Office of Naval Research (ONR). Any opinions, findings, conclusions or recommendations stated in this material are those of the authors and do not necessarily reflect the views of the funding sources.

References

1. Business email compromise: The $26 billion scam (2019), https://www.ic3.gov/media/2019/190910.aspx
2. Email security gateways. (2020), https://www.expertinsights.com/insights/top-11-email-security-gateways/
3. Multi-factor authentication (2020), https://en.wikipedia.org/wiki/Multi-factor_authentication
4. Spear-phishing email reports. (2020), https://www.phishingbox.com/
5. Aggarwal, S., Kumar, V., Sudarsan, S.: Identification and detection of phishing emails using natural language processing techniques. In: Proceedings of the 7th International Conference on Security of Information and Networks. ACM (2014)
6. Callas, J., Donnerhacke, L., Finney, H., Thayer, R.: Openpgp message format. Tech. rep., RFC 2440, November (1998)
7. Crocker, D.: Rfc0822: Standard for the format of arpa internet text messages (1982)
8. Crocker, D., Hansen, T., Kucherawy, M.: Domainkeys identified mail (dkim) signatures. RFC6376 (2011). https://doi.org/10.17487/RFC6376, https://tools.ietf.org/html/rfc6376
9. Dalton, A., Islam, M.M., Dorr, B.J., et al.: Active defense against social engineering: The case for human language technology. In: Proceedings on Social Threats in Online Conversations: Understanding and Management. pp. 1–8 (2020)
10. Duman, S., Kalkan, K., Egele, M., Robertson, W., Kirda, E.: Emailprofiler: Spearphishing filtering with header and stylometric features of emails. In: IEEE 40th COMPSAC. vol. 1, pp. 408–416. IEEE (2016)

11. Gascon, H., Ullrich, S., Stritter, B., Rieck, K.: Reading between the lines: content-agnostic detection of spear-phishing emails. In: International Symposium on Research in Attacks, Intrusions, and Defenses. pp. 69–91. Springer (2018)
12. Ho, G., Cidon, A., Gavish, L., Schweighauser, M., Paxson, V., Savage, S., Voelker, G.M., Wagner, D.: Detecting and characterizing lateral phishing at scale. In: 28th {USENIX} Security Symposium ({USENIX} Security 19). pp. 1273–1290 (2019)
13. Ho, G., Sharma, A., Javed, M., Paxson, V., Wagner, D.: Detecting credential spearphishing in enterprise settings. In: 26th {USENIX} Security Symposium ({USENIX} Security 17). pp. 469–485 (2017)
14. Hu, H., Wang, G.: End-to-end measurements of email spoofing attacks. In: 27th {USENIX} Security Symposium ({USENIX} Security 18). pp. 1095–1112 (2018)
15. Hu, X., Li, B., Zhang, Y., Zhou, C., Ma, H.: Detecting compromised email accounts from the perspective of graph topology. In: Proceedings of the 11th International Conference on Future Internet Technologies. pp. 76–82 (2016)
16. Islam, M.M., Al-Shaer, E.: Active deception framework: An extensible development environment for adaptive cyber deception. In: 2020 IEEE Cybersecurity Development (SecDev). IEEE (2020)
17. Islam, M.M., Duan, Q., Al-Shaer, E.: Specification-driven moving target defense synthesis. In: Proceedings of the 6th ACM Workshop on Moving Target Defense. pp. 13–24 (2019)
18. Khonji, M., Iraqi, Y., Andrew, J.: Mitigation of spear phishing attacks: A content-based authorship identification framework. In: 2011 International Conference for ITST. pp. 416–421. IEEE (2011)
19. Kitterman, S.: Sender policy framework (spf). RFC7208 (2014), https://tools.ietf.org/html/rfc7208
20. Klensin, J., et al.: Simple mail transfer protocol. Tech. rep., rfc 2821, April (2001)
21. Kucherawy, M., Zwicky, E.: Domain-based message authentication, reporting, and conformance (dmarc). RFC7489 (2015), https://tools.ietf.org/html/rfc7489
22. Larsen, K.G., Pettersson, P., Yi, W.: Uppaal in a nutshell. International journal on software tools for technology transfer **1**(1-2), 134–152 (1997)
23. Müller, J., Brinkmann, M., Böck, H., Schinzel, S., Schwenk, J., et al.: johnny, you are fired!–spoofing openpgp and s/mime signatures in emails. In: 28th {USENIX} Security Symposium ({USENIX} Security 19). pp. 1011–1028 (2019)
24. Ramsdell, B., et al.: S/mime version 3 message specification. Tech. rep., RFC 2633, June (1999)
25. Ruoti, S., Andersen, J., Seamons, K., et al.: "we're on the same page" a usability study of secure email using pairs of novice users. In: Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems. pp. 4298–4308 (2016)
26. Sheng, S., Broderick, L., Koranda, C.A., Hyland, J.J.: Why johnny still cant encrypt: evaluating the usability of email encryption software. In: Symposium On Usable Privacy and Security. pp. 3–4. ACM (2006)
27. Stringhini, G., Thonnard, O.: That aint you: Blocking spearphishing through behavioral modelling. In: Int. Conf. on Detection of Intrusions and Malware, and Vulnerability Assessment. pp. 78–97. Springer (2015)
28. Thomson, I.: Who's using 2fa? sweet fa. less than 10% of gmail users enable two-factor authentication. The Register (2018)
29. Verizon: 2018 data breach investigations report (2018), https://enterprise.verizon.com/resources/reports/DBIR_2018_Report_execsummary.pdf