Contents lists available at ScienceDirect

Computers & Security

journal homepage: www.elsevier.com/locate/cose

Full Length Article

SoK: The past decade of user deception in emails and today's email clients' susceptibility to phishing techniques

Maxime Fabian Veit[a,*], Oliver Wiese[b], Fabian Lucas Ballreich[a], Melanie Volkamer[a], Douglas Engels[c], Peter Mayer[d,a]

[a] Karlsruhe Institute of Technology, Germany
[b] CISPA Helmholtz Center for Information Security, Germany
[c] Freie Universität Berlin, Germany
[d] University of Southern Denmark, Denmark

ARTICLE INFO

Keywords:
Email
Email clients
Deception Techniques
Phishing Attacks
Human-Computer Interaction
Secure Mobile User Interfaces
Systemization of Knowledge

ABSTRACT

User deception in emails is still one of the biggest security risks companies and end-users face alike. Attackers try to mislead their victims when assessing whether emails are dangerous to interact with, e.g., by using techniques based on dangerous links, dangerous attachments, or both. In this work, we present a systematic literature research of deception techniques discussed in the scientific literature of the last decade. We systematize the deception techniques, focusing on techniques that use misleading sender, link, and/or attachment information. We identify 23 deception techniques which we classify as either those that email clients should protect users against (13) and those that email clients cannot protect against and thus should be addressed in security awareness measures (10). We propose a security rating for the susceptibility of email clients to these 13 deception techniques and perform an empirical evaluation to analyze the susceptibility of seven representative email clients (web, mobile apps, desktop apps) to these deception techniques. The results of our evaluation indicate that most email clients are in need of improvement to defend against the deception techniques. Hardening email clients against these deception techniques is necessary to increase the resistance against them — without unnecessarily burdening users.

1. Introduction

Email has often been called dead or in decline due to the increased use of messaging services. This, however, is not supported by usage statistics: for example, usage at one university increased significantly again during the COVID-19 pandemic (Stransky et al., 2022). Similarly, market analysts predict an increase not only in emails sent every day but also in total email users (The Radicati Group and Inc., 2021). At the same time, email is a common vector for phishing, ransomware and malware delivery (Simoiu et al., 2020). Many of these attacks deceive users about the origin of the email, the target of links, and the nature of attachments (e.g., document vs. executable). We call techniques used to deceive the user *deception techniques*. In recent years, phishing attacks have evolved to include the use of generative artificial intelligence to create more personalized phishing emails, as discussed by Shibli et al. (2024), while still employing the same repertoire of deception techniques. Email follows a decentralized approach, where different servers communicate with each other and, in addition, each user can choose from a plethora of different email clients (e.g., Apple Mail, Outlook, Gmail...). The flexibility afforded by the decentralized structure of email has allowed it to grow into a complex and modular system with a multitude of specifications. Yet, not all specifications are supported by all applications and servers. A client cannot expect the server to be able to detect and prevent all deception techniques. Consequently, following usable security by design best practices, email clients should be able to detect deception techniques in emails whenever possible and either prevent them or support users in detecting them. A multitude of possible deception techniques in the context of email have been published (e.g., Suriya et al., 2009; Alazab and Broadhurst, 2016; Hannay and Bolan, 2009). Yet, a clear systematization and overview of published deception techniques is missing.

Therefore, the first goal of our work is to provide the missing systematization and overview. Additionally, our second goal is to study how current email clients handle these known attacks, i.e., whether they help users protect themselves or leave them susceptible. These two

* Corresponding author.
E-mail addresses: maxime.veit@kit.edu (M.F. Veit), oliver.wiese@cispa.de (O. Wiese), fabian.ballreich@kit.edu (F.L. Ballreich), melanie.volkamer@kit.edu (M. Volkamer), douglas.engels@fu-berlin.de (D. Engels), mayer@imada.sdu.dk (P. Mayer).

https://doi.org/10.1016/j.cose.2024.104197
Received 29 April 2024; Received in revised form 26 September 2024; Accepted 5 November 2024
Available online 28 November 2024

goals are captured in the research questions that we aim to answer in this work:

RQ 1 Which deception techniques have been reported in the literature of the past decade that aim at deceiving users about the identity of senders, the targets of links, or the nature of attachments?

RQ 2 Are common email clients susceptible to these deception techniques?

To answer our first research question, we conduct a systematic literature review of deception techniques in emails of the past decade. In it, we find overall 23 different deception techniques, that cover all three of (a) deceiving users about the identity of senders, (b) deceiving users about the targets of links, and (c) deceiving users about the nature of attachments.

To answer the second research question, we then classify these deception techniques distinguishing whether detection of the deception technique would require additional information, finding 13 of these deception techniques would not require additional information, and email clients should therefore be able to block them or offer support to users. We then investigate the susceptibility of email clients to different instantiations of these 13 deception techniques. We find that most of these deception techniques are viable and could deceive users in several different email clients. Through a classification using our security rating, we find that 38.1% of tested cases should be rated as highly susceptible. Deception techniques based on links seem to fare worst in this regard with 41.3% of cases being high.

In summary, the main contributions are:

- We conducted a systematic literature review and present a catalogue of the deception techniques in emails identified in the literature of the past decade (Section 3).
- We evaluate the identified deception techniques and present a classification of them, distinguishing those deception techniques for which email clients can offer support without external information (Section 4).
- We contextualize the deception techniques by evaluating the susceptibility of a representative set of email clients to instantiations of these deception techniques and derive a security rating as suitable taxonomy from the analysis (Section 5).
- We provide (a) EML files of the deception technique instantiations of our analysis and (b) the individual results for each email client as screenshots publicly in a GitHub repository.[1] Thereby, we aim to increase the replicability of our work and provide developers of email clients with tools to check the susceptibility of their clients to these deception techniques.

2. Background and preconsiderations

In this section, we provide some background information and the necessary definitions, and conclude by narrowing down what is not the focus of this work.

2.1. Scope of this work

Phishing is a widely used term, but is used for different scopes of attacks. The Anti-Phishing Working Group's definition focuses on *stealing consumers' personal identity data and financial account credentials* by means of social engineering (e.g., deceptive email addresses) or technical subterfuge (e.g., malware) as goal of the attacker (Group et al., 2005). However, other attacker goals are described for phishing in the literature, e.g., in business email compromise the attacker tries to get the victim to perform a certain action such as wiring money (potentially without stealing credentials) (Cross and Gillett, 2020). Other additional attacker goals include, e.g., *planting ransomware* on the victims' devices (also potentially without stealing credentials) (Thomas, 2018).

In this work, we do not focus on a specific attacker goal, but rather investigate techniques that attackers could use (possibly multiple in combination) in email-based phishing attacks to reach their goals. Thereby, the goal of our work is to highlight ways in which email clients can be improved to support users better in detecting deceptive techniques used in phishing attacks. To that end, we investigate how attackers can attempt to deceive users by exploiting how email clients display different elements of messages. We specifically focus on three distinct elements: (a) the target URL of a *link*, (b) the email address of the *sender*, and (c) the file extension of the *attachment*. We include the sender in our considered security indicators for several reasons. First, many security awareness measures emphasize the importance of verifying sender information (see Reinheimer et al., 2020). If a user is successfully misled about the sender's identity, they may become more susceptible to further attacks within the same email, such as clicking on malicious links or opening harmful attachments without proper caution. While it might be argued that a properly configured Sender Policy Framework (SPF) could mitigate this risk, our analysis of deception techniques (see Section 3.2) shows that SPF only addresses one of the ten identified deception techniques related to the sender field. Moreover, a recent study found that 68% of 2.2 billion scanned domains do not use SPF (Sangwan, 2022), highlighting the problem.

2.2. Differences to related work

To the best of our knowledge, two other related systematization works in the context of phishing exist. Firstly, Franz et al. (2021) systematized user-oriented interventions along the four categories Education, Training, Awareness-raising, and Design. From this systematization, they derive recommendations for future phishing interventions. Thus, they focus on the defense against phishing attacks by users through interventions.

Secondly, Zhuo et al. (2023) systematized phishing user studies. They investigated which factors for phishing susceptibility were researched in these user studies, e.g., stress, personal knowledge, mental fatigue, and emotional reactions by the users. From their review of these studies they derived their Phishing Susceptibility Model which systematizes these factors along two axis: (a) whether they are stable or fluctuating and (b) whether they are short-term or long-term. Thus they focus on the individual user's susceptibility to phishing attacks.

While both of these works point out the importance of phishing detection in the context of email, different attack and (technical) deception strategies were out of scope for both. Thus, our systematization of deception techniques complements these related works with a more technical view on techniques to deceive users based on an email's sender, links, or attachments. Specifically, we focus on the susceptibility of email clients to the deception techniques we found and how well they support users in the detection of deception techniques.

2.3. Terminology

In this section, we introduce relevant terminology.

2.3.1. Target URLs of links

When a link is hovered, email clients (which include email web interfaces used in web browsers) on desktop/laptop devices display the target URL in the *statusbar* and/or in a *tooltip*. On mobile devices (smartphone/tablet), the process of identifying the target URL strongly depends on the device and the respective app. In most cases, it is displayed in a dialog window when touching the link accordingly. The target URL can either be trustworthy or dangerous. In the latter case, the corresponding web service is controlled by the attacker.

[1] Link will be provided in full version due to anonymity during the review.

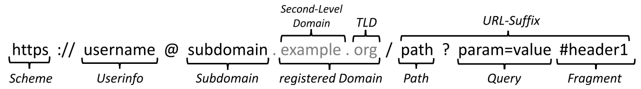


Figure 1. URL structure and nomenclature of its parts. (TLD: Top-Level-Domain).

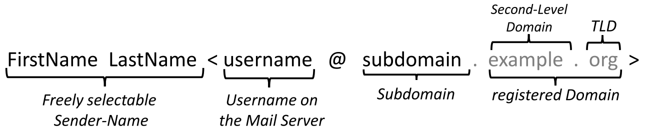


Figure 2. Sender structure and terminology for different parts. (TLD: Top-Level-Domain).

2.3.2. URL structure

URLs follow a specific structure, shown in Fig. 1. The most relevant part of a URL is the *registered domain*. The registered domain information is the one that determines which web service is accessed once the link is clicked.

2.3.3. Sender address

The structure of a sender address is similar to that of a URL, although simpler (see Fig. 2). As before, the most relevant information is the registered domain. The sender address is part of the email header.

2.3.4. File extension

Every file is defined by its name and extension, divided by a dot. The file name is not limited to letters and numbers, but allows other characters, e.g., "." and "-". Some file extensions can be considered more or less dangerous than others. Executable files (e.g., .exe files on Windows) are the most critical ones.

2.4. Out of scope

Attackers can further mislead victims when encrypting and/or signing dangerous emails (using either PGP or S/MIME). While a few publications consider such attacks, e.g. Müller et al. (2019a), we do not consider attacks based on OpenGPG and S/MIME in this work. Even in organizations with existing public key infrastructures encryption and digital signatures are not widely used (Stransky et al., 2022). As there are plenty of attack vectors that work independently from using PGP and/or S/MIME, these attacks are less relevant than others.

While much work focuses on mail infrastructure attacks (Foster et al., 2015; Hu and Wang, 2018; Hu et al., 2018; Clark et al., 2021; Chen et al., 2020), we focus on user deception in email clients.

We also do not consider deception techniques integrating JavaScript in an email for two reasons: (a) The vast majority of email clients block JavaScript by default[2]; and (b) if an email client does not block JavaScript, an attacker would be able to change the email behaviors in arbitrary ways due to the power of JavaScript. This would allow a malicious website to be opened without the user having to be deceived first, which is the main aim of link deception techniques. Yet, there are publications discussing such deception techniques, e.g. Heiderich et al. (2011), Khonji et al. (2011).

3. Literature review

In order to systematize deception techniques the method of systematic literature research (Kitchenham, 2004) is used. Our procedure as well as the results are discussed in this section.

3.1. Methodology

In our systematic literature review, we considered only publications published in the last decade before the literature review (i.e., from 2010 – 2021, inclusive), in order to not include outdated findings. As visualized in Fig. 3, we used two approaches to collect the publications:

(a) the five most widely recognized databases of scientific literature in the computer science field, i.e., ScienceDirect, ACM, IEEE Xplore, SpringerLink, and Scopus; and
(b) conferences and journals not indexed by the aforementioned databases but publishing relevant related work, i.e., Usenix Security Symposium, Symposium On Usable Privacy and Security, Usable Security Workshop.

We queried these sources with three search terms: "email security", "email vulnerability", and "email attack". To be selected, either the title or the abstract had to contain one of the terms.

We found 2224 publications with the search terms and reduced the number of relevant literature by applying additional criteria:

- We excluded publications behind "paywalls" (i.e., not licensed by the authors' university library) and/or non-peer-reviewed publications (e.g., editorials, position papers, etc.).
- We then filtered out those publications not describing deception techniques focused on emails, by manually scanning the title, abstract, and if necessary, the full text.

For the latter, the literature was assessed by two authors using a 5-point scale from not relevant to highly relevant, first for the title and abstract and second for the full text.[3] In advance, the authors agreed on the definition of deception techniques and the focus (see Section 2.1). Borderline cases and ambiguities were discussed along with the other authors.

Overall, 99 publications met these additional criteria. To broaden our results, we performed first a forward, then a backward search, followed by the manual scanning described above, resulting in a final number of overall 111 relevant publications.

In the final step, we read these 111 publications reporting deception techniques that focus on emails to select those describing deception techniques targeting the sender field, links, and/or attachments.

3.2. Results - Overview

From the 111 publications, we identified 47 publications describing deception techniques. We categorized them into a total of 23 unique deception techniques that actually target the sender field, links, and/or attachments. Several publications contained multiple deception techniques. Some deception techniques were reported by multiple publications, while others were reported for two of the considered indicators (e.g., both for the sender and the link). The categorization process underwent mutual cross-checks among the authors to enhance objectivity and the procedure was documented.[3] Of these 23 deception techniques, four target the sender, 18 the link, and four the attachment. Three of them target two different indicators. Table 1 gives an overview of all 23 deception techniques and which indicators they target.

3.3. Results – Deception Technique descriptions

In the following, we describe the 23 deception techniques in detail and discuss if deception techniques are applicable to further indicators.

[2] Javascript support of email clients, https://en.wikipedia.org/wiki/Comparison_of_email_clients#Messages_features.

[3] Additional documentation of the evaluation of the literature - https://github.com/SecUSo/paper-artifacts-email-deception-techniques/blob/main/1.LiteratureResearch/Literature_Combined_All_v0.6.2.xlsx.

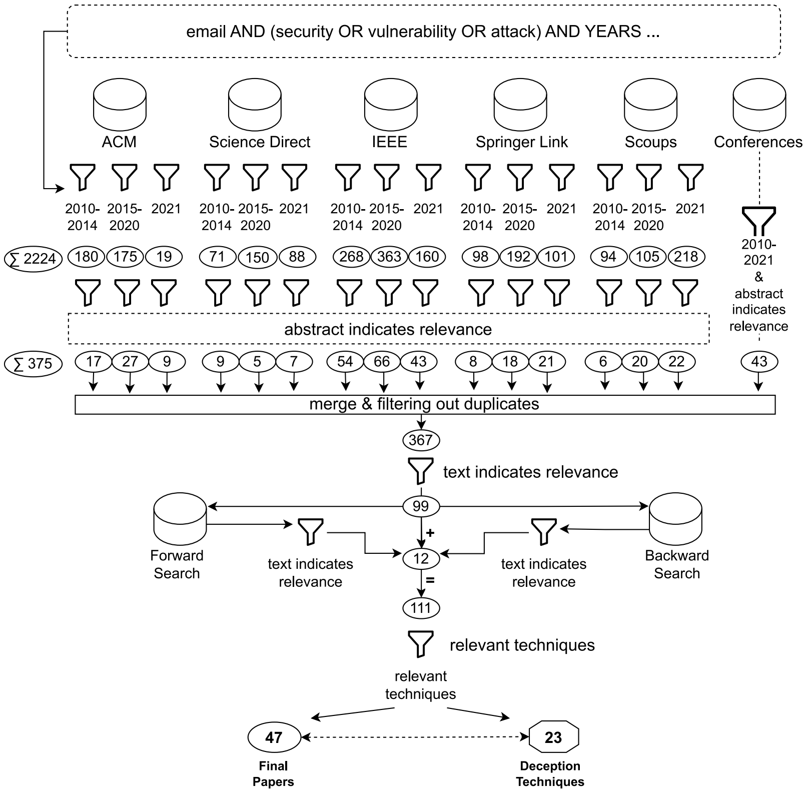


Figure 3. Procedure systematic literature research.

3.3.1. Sender-Mail Spoofing

In this deception technique the attacker's goal is to pretend to be someone else by spoofing the sender email address (Pirocca et al., 2020; Maroofi et al., 2021; Soussi et al., 2020; Du and Xue, 2013; Weaver et al., 2021; Jakobsson, 2016; Shen et al., 2021). As the sender field of the email is determined on the sender side, the attacker can simply change it. A proper configuration of SPF for both the spoofed domain and the receiving mail server would prevent this deception technique.

3.3.2. Sender-Name Spoofing

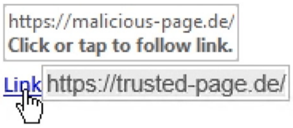
For this deception technique, the attacker's goal is to pretend to be someone else by showing a *fake sender name* in the sender field (Jakobsson, 2016). The content of the sender field is determined on the sender side (see sender field structure in Section 2.3.3). Thus, the attacker can easily change it. The danger of this deception technique is that several email clients display by default only the sender name, without the sender email address. An advanced version of this deception technique is to use an email address as sender name (e.g., the name is order@amazon.com).

3.3.3. Fake Tooltip

For the deception technique mentioned in Reinheimer et al. (2020) the attacker shows a (fake) tooltip with a URL defined by themselves (most likely a trustworthy one) once their victims hover with the mouse over a link (see Fig. 4). The URL displayed in the fake tooltip is not the actual target URL of this link (compare real tooltip described in Section 2.3.1). There are different ways for attackers to integrate such tooltips in emails, e.g., using CSS and the HTML title-attribute, but (Reinheimer et al., 2020) does not specify this.

3.3.4. Hexadecimal Encoding

This deception technique (Sankhwar et al., 2018; Du and Xue, 2013) exploits hexadecimal encoded characters in URLs of links, in particular in (sub)domains. However, it is also possible to use hexadecimal encoded characters for some of the characters in the domain.



https://malicious-page.de/
Click or tap to follow link.

Link https://trusted-page.de/

Figure 4. Example of "Fake Tooltip". Displays both, fake (trusted-page.de) and original (malicious-page.de) tooltip as full-url at the same time.

For example, an attacker would register *demal.is*, would encode it to *de%6D%61%6C%2E%69%73*, and would add Google as a subdomain. The target URL of the corresponding link would then contain *google.de%6D%61%6C%2E%69%73*. Hexadecimal encoding is not easily human-readable and it becomes more difficult for users to identify the relevant security indicators of the URL.

3.3.5. HTML Base-Tag

For this deception technique attackers use the HTML base-tag before links (Orman, 2012). It specifies the base URL for all relative URLs in the HTML email. E.g., the base part https://malicious-page.de/ is extended using the relative part *trusted-page.de* (see Fig. 5). If relative URLs are used after the *HTML base-tag* the absolute part specified by the *HTML base-tag* is appended as a prefix. This deception technique assumes that when hovering over the corresponding link in an email client only the relative part of the URL is displayed (either in the statusbar and/or the tooltip depending on the email client).

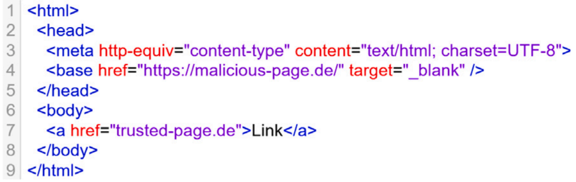
3.3.6. HTML Form-Tag

In this deception technique, the attacker takes advantage of HTML forms to hide the target URL of links, as the latter is usually not shown in the tooltip or the status bar when hovering over the form's submit button (Qabajeh and Thabtah, 2014; Du and Xue, 2013). The target URL would then be the attacker's dangerous web service (example in Fig. 6). The attacker can make the submission button look like a

Table 1
Deception techniques found in the literature on sender, link, and attachment and our consideration if they can also be applied to other indicators (visualized by *). The last column is based on the classification presented in Section 4 and shows which deception techniques were considered for the analysis in Section 5. (Link to online resource to all instantiations of deception techniques).

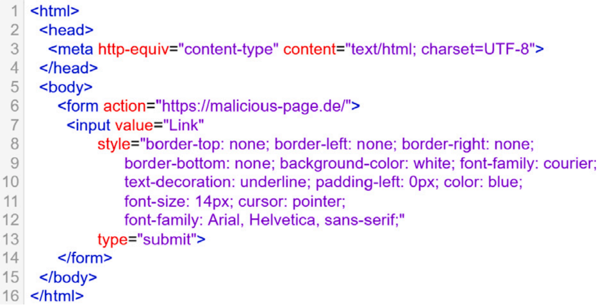
Deception Technique	Sender	Link	Attachment	Instantiation
(1) Sender-Mail Spoofing	Pirocca et al. (2020), Maroofi et al. (2021), Soussi et al. (2020), Du and Xue (2013), Weaver et al. (2021), Jakobsson (2016), Shen et al. (2021)			No
(2) Sender-Name Spoofing	Jakobsson (2016)			Yes
(3) Fake Tooltip		Reinheimer et al. (2020)		Yes
(4) Hexadecimal Encoding		Sankhwar et al. (2018), Du and Xue (2013)		Yes
(5) HTML Base-Tag		Orman (2012)		Yes
(6) HTML Form-Tag		Qabajeh and Thabtah (2014), Du and Xue (2013)		Yes
(7) Link Mismatch		Pearson et al. (2017), Sankhwar et al. (2018), Suriya et al. (2009), Li et al. (2020), Khursheed et al. (2020)		Yes
(8) Mailto-Scheme		Müller et al. (2020)		Yes
(9) Redirect		Volkamer et al. (2017)		No
(10) Ropemaker		Müller et al. (2019b)		Yes
(11) URL Posing-Suffix		Soni et al. (2011), Jampen et al. (2020a)		No
(12) URL Shortener		Kokubun and Nakamura (2018), Gupta et al. (2014), Pithawala et al. (2021), Lee et al. (2021), Zhu et al. (2020), Awasthi and Goel (2021), Priya et al. (2020), Suriya et al. (2009), Abedin et al. (2020), Salahdine et al. (2021), Singh and Meenu (2020), Swarnalatha et al. (2021), Du and Xue (2013), Alazab and Broadhurst (2016), Bhardwaj et al. (2020)		No
(13) URL Userinfo Field		Sankhwar et al. (2018), Qabajeh and Thabtah (2014), Suriya et al. (2009), Vrbančič et al. (2020), Zhu et al. (2020), Awasthi and Goel (2021), Priya et al. (2020), Azeez et al. (2020), Abedin et al. (2020), Salloum et al. (2021), Singh and Meenu (2020), Swarnalatha et al. (2021), Du and Xue (2013), Bhardwaj et al. (2020)		Yes
(14) Double File-Extension			Alazab and Broadhurst (2016)	Yes
(15) Unknown File Extension			Balan and Popescu (2018), Alazab and Broadhurst (2016)	No
(16) Homographic Spoofing	Shen et al. (2021)	Suriya et al. (2009), Hu et al. (2021a), Hannay and Bolan (2009), Andryukhin (2019), Du and Xue (2013), Wang et al. (2006)		Yes
(17) Right-to-Left-Override	*		Marczak et al. (2014), Alazab and Broadhurst (2016)	Yes
(18) Unrecognizable Domain	*	Sankhwar et al. (2018), Soni et al. (2011), Qabajeh and Thabtah (2014), Suriya et al. (2009), Vrbančič et al. (2020), Li et al. (2020), Lee et al. (2021), Jampen et al. (2020a), Zhu et al. (2020), Awasthi and Goel (2021), Priya et al. (2020), Azeez et al. (2020), Abedin et al. (2020), Salahdine et al. (2021), Salloum et al. (2021), Singh and Meenu (2020), Swarnalatha et al. (2021), Du and Xue (2013), Volkamer et al. (2017), Bhardwaj et al. (2020)		No
(19) Domain Extension	*	Sankhwar et al. (2018), Suriya et al. (2009), Jampen et al. (2020b), Awasthi and Goel (2021), Azeez et al. (2020), Abedin et al. (2020), Bhardwaj et al. (2020)		No
(20) Subdomain	*	T. N et al. (2021), Jampen et al. (2020a), Blum et al. (2010), Abedin et al. (2020), Swarnalatha et al. (2021), Bhardwaj et al. (2020)		No
(21) Different TLD	*	T. N et al. (2021)		No
(22) Mangle	Jakobsson (2016), Soni et al. (2011), Sankhwar et al. (2018), Wang et al. (2006), Agten et al. (2015), Pearson et al. (2017), Spaulding et al. (2016), Suriya et al. (2009), T. N et al. (2021), Pithawala et al. (2021), Jampen et al. (2020a), Pirocca et al. (2020), Volkamer et al. (2017)	Jakobsson (2016)	*	No
(23) Exceedingly Long	*	Sankhwar et al. (2018), Soni et al. (2011), Suriya et al. (2009), Zhu et al. (2020), Priya et al. (2020), Azeez et al. (2020), Abedin et al. (2020), Singh and Meenu (2020), Swarnalatha et al. (2021), Bhardwaj et al. (2020)	Alazab and Broadhurst (2016)	Yes

*=Extendable to this indicator.



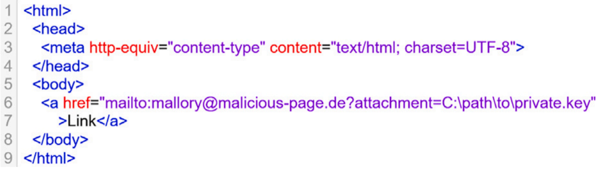
```
<html>
 <head>
  <meta http-equiv="content-type" content="text/html; charset=UTF-8">
  <base href="https://malicious-page.de/" target="_blank" />
 </head>
 <body>
  <a href="trusted-page.de">Link</a>
 </body>
</html>
```

Figure 5. Example code for deception technique "HTML Base-Tag".



```
<html>
 <head>
  <meta http-equiv="content-type" content="text/html; charset=UTF-8">
 </head>
 <body>
   <form action="https://malicious-page.de/">
    <input value="Link"
        style="border-top: none; border-left: none; border-right: none;
           border-bottom: none; background-color: white; font-family: courier;
           text-decoration: underline; padding-left: 0px; color: blue;
           font-size: 14px; cursor: pointer;
           font-family: Arial, Helvetica, sans-serif;"
        type="submit">
   </form>
 </body>
</html>
```

Figure 6. Example code for deception technique "HTML Form-Tag".



```
<html>
 <head>
  <meta http-equiv="content-type" content="text/html; charset=UTF-8">
 </head>
 <body>
  <a href="mailto:mallory@malicious-page.de?attachment=C:\path\to\private.key"
    >Link</a>
 </body>
</html>
```

Figure 7. Example code for deception technique "Mailto-Scheme".

normal looking "click here" link or URL-link by using inline CSS. This deception technique gets more difficult to detect if it is combined with the "Fake Tooltip" deception technique (cf. Section 3.3.3).

3.3.7. *Link Mismatch*

For this deception technique, the attacker exploits that the link text does not need to match the target URL (Pearson et al., 2017; Sankhwar et al., 2018; Suriya et al., 2009; Li et al., 2020; Khursheed et al., 2020). The attacker thus chooses a trustworthy-looking URL (e.g., amazon.com) as link text, but the actual target URL directs the victims to a web service controlled by the attacker.

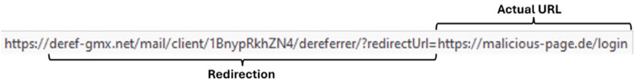
3.3.8. *Mailto-Scheme*

In this deception technique, the attacker uses the "mailto" scheme in links to get the user to send sensitive data (like a private key) as an attachment to the attacker (Müller et al., 2020).

Using the "mailto" scheme in the link, not only a receiver email can be determined, but also the email text, body, and (more importantly for this deception technique) some email clients allow local files to be specified as attachments (see Fig. 7). Once the link is clicked, an email to the attacker containing sensitive data as an attachment is prepared. However, a successful deception technique is possible only if the victim then sends the email.

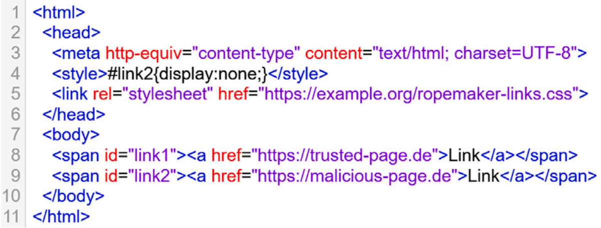
3.3.9. *Redirect*

This deception technique takes advantage of redirect services for links (Volkamer et al., 2017). These services allow anyone to create URLs using the domain name of this legitimate service, redirecting to a pre-defined URL, e.g., https://trusted-page.de/deref?q=https://malicious-page.de redirects to https://malicious-page.de. The actual purposes of redirect services are manifold: (a) for marketing reasons, to track which links are clicked, e.g., in newsletters; (b) for privacy reasons, to prevent the targeted web service to know who sent the request; (c) for security or privacy reasons, to check the URL before it is opened or removing the URL of the referring page (see Fig. 8). There are also unintended "open" redirect services, caused by misconfigurations or vulnerabilities of servers (Shue et al., 2008). Depending on the implementation of the redirect service, the pre-defined URL can either be part of the path or not. In both cases, checking the domain of the target URL is not sufficient to decide how risky it is to click on such links. Note further, even if the target URL is part of the path these might still be problematic due to issues with displaying long URLs in email clients (cf. Section 3.3.23).



https://deref-gmx.net/mail/client/1BnypRkhZN4/dereferrer/?redirectUrl=https://malicious-page.de/login

Figure 8. Legitimate redirects by an email provider are misused here to disguise the destination *malicious-page.de*. Users who follow common security awareness training (Reinheimer et al., 2020) may only check the *deeref-gmx.net* part of the URL, unaware that it actually redirects to *malicious-page.de*.



```
<html>
 <head>
  <meta http-equiv="content-type" content="text/html; charset=UTF-8">
  <style>#link2{display:none;}</style>
  <link rel="stylesheet" href="https://example.org/ropemaker-links.css">
 </head>
 <body>
  <span id="link1"><a href="https://trusted-page.de">Link</a></span>
  <span id="link2"><a href="https://malicious-page.de">Link</a></span>
 </body>
</html>
```

Figure 9. Example code of deception technique "Ropemaker".

3.3.10. *Ropemaker*

In this deception technique, the attacker's goal is to (unexpectedly) change a target URL of a link (Müller et al., 2019b). For example, the first time an email is opened, the target URL of a link is a trustworthy one. When the email is opened again, the target URL directs to the attacker's server. One potential scenario is that the receiver opens the email, checks the link, judges if its safe to be clicked, and forwards the email contacts. These contacts may not check the link carefully because they trust the initial receiver. However, the target URL now directs to the attacker's web service. For this deception technique, the email embeds remote CSS (see Fig. 9).

3.3.11. *URL Posing-Suffix*

For this deception technique, attackers exploit that adding a string in the URL path, in the query part, or in the fragment part of a URL does not influence which web server is accessed (Soni et al., 2011; Jampen et al., 2020a) (See URL structure described in Section 2.3.2). Hence, the attacker can register any domain they want and then add a link to the email with a target URL to this server. However, the target URL would also contain a trustworthy URL (or parts of it) in one of the three URL parts mentioned before, e.g., https://connection.io/google.com/login directs victims to *connection.io* even if it contains *google.com/login*. The attacker try to impersonate a redirect service to further confuse users, e.g., using https://connection.io/redirect=https://www.google.com/login.

3.3.12. *URL Shortener*

In this deception technique, the attacker's goal is to hide the actual target URL of the link in a URL shortening service, which is then displayed as the target URL in a tooltip or a status bar (Kokubun and Nakamura, 2018; Gupta et al., 2014; Pithawala et al., 2021; Lee et al., 2021; Zhu et al., 2020; Awasthi and Goel, 2021; Priya et al., 2020; Suriya et al., 2009; Abedin et al., 2020; Salahdine et al., 2021;

Singh and Meenu, 2020; Swarnalatha et al., 2021; Du and Xue, 2013; Alazab and Broadhurst, 2016; Bhardwaj et al., 2020). The shortened URL redirects the victim to the attacker's web service. Attackers can easily obtain a shortened URL because these services are often free and do not require registration.

3.3.13. URL Userinfo Field

This deception technique makes use of the *userinfo part* of the target URL (Sankhwar et al., 2018; Qabajeh and Thabtah, 2014; Suriya et al., 2009; Vrbančič et al., 2020; Zhu et al., 2020; Awasthi and Goel, 2021; Priya et al., 2020; Azeez et al., 2020; Abedin et al., 2020; Salloum et al., 2021; Singh and Meenu, 2020; Swarnalatha et al., 2021; Du and Xue, 2013; Bhardwaj et al., 2020) (See URL structure described in Section 2.3.2). This optional part of a URL can be used to authenticate users on websites. The content in this part can be freely chosen by the attacker, leading to attack URLs like https://google.com@connection.io/ which direct victims to *connection.io*.

3.3.14. Double File-Extension

In this deception technique, a dangerous executable file (e.g., *.exe*) is attached to an email (Alazab and Broadhurst, 2016). As the file extension reveals that it is an executable file, the attacker uses a file name that looks like a low-risk file type, e.g., *cv.pdf.exe*. This deception technique is particularly difficult to detect if email clients do not display the file extension. As only the file type determines how the attachment is opened, once clicked, the file (in case of an attack, the malware) would be executed. Note, while the publication talks about a double file-extension, the number of file-extensions is arbitrary, i.e., the attacker can use more than two, e.g., *cv.pdf.no.exe*.

3.3.15. Unknown File Extension

In this deception technique, the attackers assume users will recognize file extensions of the dangerous attachments (Balan and Popescu, 2018; Alazab and Broadhurst, 2016) (See Section 2.3.4). While many users may know that ".exe" files are dangerous, they may not know other dangerous file types executed in Windows OS, such as *.pif*, *.scr*, or *.com*. The attacker can also exploit confusing file extensions, e.g., *.scr* could be interpreted as *scr*eenshots or *.com* may be considered a *top-level domain*.

3.3.16. Homographic Spoofing

This deception technique is mentioned in Shen et al. (2021) for the "sender" indicator and in Suriya et al. (2009), Hu et al. (2021a), Hannay and Bolan (2009), Andryukhin (2019), Du and Xue (2013), Wang et al. (2006) for the "link" indicator. In it, the attacker registers a new domain which is homographically equal or similar to the targeted brand.

It is possible to use letters from different alphabets as domain names. Non-ASCII characters, i.e., Unicode characters, are encoded in Punycode for the domain registration (Klensin, 2010), but can be displayed in decoded form. With Homographic Spoofing, this is exploited by mixing alphabets that contain similar-looking, i.e. homographically identical, letters. This makes it possible, for example, to use the Cyrillic alphabet, which contains characters that look exactly like some Latin characters (e.g., у instead of y) in a domain name. Thus, the attacker can register domains that look similar to the one in Latin characters but are actually different ones (e.g., уаһоо.com instead of yahoo.com) which can be encoded using code (e.g., *xn–80a2aar51d.com* for уаһоо.com). Hence, depending on whether an email client displays the domain as punycode or not, corresponding deception techniques are more or less easily detectable. The important trade-off is allowing Non-ASCII characters in contexts where it makes sense. As outlined above, the most critical case is when different alphabets are mixed. In theory, this deception technique can be extended to the "attachment" indicator, but to the best of our knowledge there are no , i.e. executable, file extensions with non-ASCII characters.

3.3.17. Right-to-Left-Override

This deception technique uses the Unicode RLO-character "U+202e" as part of the attachment file name (Alazab and Broadhurst, 2016; Marczak et al., 2014). This Unicode character changes the writing direction of the text so that characters are displayed right-to-left instead of left-to-right. For example, the file name *document-fdp.exe*, written as *document-[RLO-character]fdp.exe*, would be displayed as *document-exe.pdf*.

As indicated with "*" in Table 1, deception technique can be extended to the sender by reversing the sender domain address and thereby bypass SPF, e.g., *alice@[RLO-character]moc.elgoog.com* would be displayed as alice@moc.google.com.

3.3.18. Unrecognizable Domain

In this deception technique, the attacker's goal is to use as target URL either an IP address or a domain that is essentially an unrecognizable string (e.g., aldslkskmskoewlc.at) for the URL in a link (Sankhwar et al., 2018; Soni et al., 2011; Qabajeh and Thabtah, 2014; Suriya et al., 2009; Vrbančič et al., 2020; Li et al., 2020; Lee et al., 2021; Jampen et al., 2020a; Zhu et al., 2020; Awasthi and Goel, 2021; Priya et al., 2020; Azeez et al., 2020; Abedin et al., 2020; Salahdine et al., 2021; Salloum et al., 2021; Singh and Meenu, 2020; Swarnalatha et al., 2021; Du and Xue, 2013; Volkamer et al., 2017; Bhardwaj et al., 2020). This deception technique exploits the fact that users cannot properly judge such URLs and may decide they are trustworthy based on the rest of the email. This deception technique can be extended to the "sender" indicator.

3.3.19. Domain Extension

For this deception technique, the attacker registers a new domain containing the targeted brand in the domain name (Sankhwar et al., 2018; Suriya et al., 2009; Jampen et al., 2020b; Awasthi and Goel, 2021; Azeez et al., 2020; Abedin et al., 2020; Bhardwaj et al., 2020), e.g., *secure-google.com* or *securegoogle.com* to imitate *google.com*. The extension could either be any term (e.g., *in-google.com*) or various types of trustworthiness ones (e.g., *google-https.com*, *safe-at-google.com*). This deception technique is based on the difficulty to determine whether such domains actually belong to the same brand as the original one.

This deception technique can be extended to the "sender" indicator using the same approach for the domain of the email address (e.g., support@secure-google.com). Attackers can thus even combine this and use the same domain for both, sender and link.

3.3.20. Subdomain

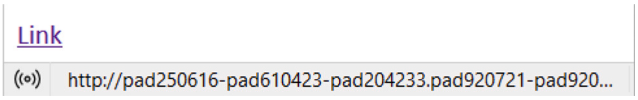
This deception technique exploits the left-to-right reading direction of most Western cultures (T. N et al., 2021; Jampen et al., 2020a; Blum et al., 2010; Abedin et al., 2020; Swarnalatha et al., 2021; Bhardwaj et al., 2020) (See URL structure described in Section 2.3.2). The domain registered by the attacker is not relevant because when pretending to have a link with a target URL to, e.g., Google, the attacker uses, e.g., google.com as subdomain (e.g., google.com.connection.io).

This deception technique can be extended to the "sender" indicator by using subdomains for the email address, e.g., noreply@amazon.de.connection.io.

3.3.21. Different TLD

For this deception technique, the attacker registers a new domain using the targeted brand but with a different top-level domain (TLD) (T. N et al., 2021). The attacker exploits that there are too many TLDs for a brand to register them all, e.g., for *youtube.com*, the attacker could register *youtube.cc*. Furthermore, it is difficult to know which of the TLDs have been registered by a brand.

This deception technique can be extended to the "sender" indicator by using, e.g., support@youtube.cc instead of support@youtube.com.



Link

http://pad250616-pad610423-pad204233.pad920721-pad920...

Figure 10. Example for Deception technique "Exceedingly Long". Screenshot from Mozilla Thunderbird.

3.3.22. Mangle

This deception technique is mentioned in Soni et al. (2011), Sankhwar et al. (2018), Wang et al. (2006), Agten et al. (2015), Pearson et al. (2017), Spaulding et al. (2016), Suriya et al. (2009), T. N et al. (2021), Pithawala et al. (2021), Jampen et al. (2020a), Pirocca et al. (2020), Volkamer et al. (2017), Jakobsson (2016) for the "link" indicator and in Jakobsson (2016) also for the sender. For this deception technique, the attacker registers a new domain with a domain name very similar to the targeted brand. It exploits that most brands cannot ensure that similar-looking domains are not registered by attackers. Examples of this deception technique are intentional typos (e.g., *mircosoft* instead of *microsoft*), replacing characters with similar looking ones (e.g., *m* with *rn*), or extensions (e.g., doubling a character or adding a -).

This deception technique can also be applied to file extensions (e.g., *.pif* instead of *.pdf*).

3.3.23. Exceedingly Long

This deception technique is mentioned in Sankhwar et al. (2018), Soni et al. (2011), Suriya et al. (2009), Zhu et al. (2020), Priya et al. (2020), Azeez et al. (2020), Abedin et al. (2020), Singh and Meenu (2020), Swarnalatha et al. (2021), Bhardwaj et al. (2020) for the "link" indicator and in Alazab and Broadhurst (2016) for the "attachment" indicator. In this deception technique, the attacker uses very long (sub)domains[4] in the URL or the file name of the attachment.[5] This makes it very difficult to identify the relevant security indicators — in particular if, due to space constraints, an email client does not display the entire URL or file name (see Fig. 10). For file names of attachments, space can also be used for padding between the file name and the file extension (e.g., *invoice.doc   .exe*).

This deception technique can also be extended to the "sender" indicator. Long domains can be registered and in addition, subdomains can be used. The sender's address can be longer than 253 characters, as it does not necessarily have to be registered.

4. Classification

The goal of the deception techniques described in the previous section is to mislead the user when assessing whether emails are dangerous to interact with or not.

Ideally, email clients would support users in detecting all of these deception techniques. However, some deception techniques can only be detected by either the email server (which has additional information, e.g., about the sending server) or the user (who can put the relevant information in context). Therefore, we distinguish between the following classes of deception techniques:

1. Client support without additional information possible (Support based on information available to the email client, e.g., URL and text of the links)
2. Client support would require additional information (e.g., which services a user is using)

In the following subsections, we explain which of the 23 deception techniques we assign to which of the two classes. As the remainder of our work will focus on the deception techniques in the class (1), we first consider the other class to make it easier for the reader to follow.

4.1. Client support would require additional information

Sender-Mail Spoofing. The server can detect this deception technique with additional methods like SPF. However, for the email client this is not possible as it has neither a connection to the sender email server nor access to the IP address of it.

Mangle, Different TLD, Subdomain, Domain Extension, URL Posing-Suffix, Unrecognizable Domain. There are several deception techniques that could be addressed by email clients if a list of trustworthy domains would exist. Yet, this is not the case. For example, using amazon.com as a subdomain or in the path can be useful for certain web services that compare different services. Also, some organizations are using different registered domains, e.g., there is lufthansa.com, lufthansa-cargo.com, and lufthansagroup.com but there is no way to find out which other domains Lufthansa has registered. Similar issues arise when the email client attempts to identify the deception technique Mangle. For instance, while *gmail.de* is a legitimate Google domain, *gmil.de* – though not associated with Google — could still be a trustworthy website belonging to another company. Email clients would need context to decide. Users may have this context and, if aware of these deception technique, may be able to detect them.

Redirect, URL Shortener. Both deception techniques are not detectable without additional information — in the worst case, by opening the link in a safe environment (e.g., Virtual Machine or Sandbox). Email clients could provide some support regarding these types, if a complete list of corresponding services would exist. However, this is not the case, as it is easy to create new ones and the landscape is dynamic in this respect.

Unknown File Extension. This deception technique could be addressed by email clients, if a list of trustworthy file extensions would exist. However, this is not the case because there are various programs that the user may have installed, and it is not clear how the programs will handle the file (e.g., execute it or not). Users may be aware of which file types they can open with low risk in which context.

4.2. Client support without additional information possible

This class contains deception techniques for which email clients can unambiguously decide if the corresponding misleading element is in an email. Having identified something to be misleading does not necessarily mean that the email is dangerous. It might be that the element was integrated by accident or without knowing that it is also used by attackers. For instance, it is common among marketing companies to redirect links through services to track email campaign reach, keeping the URL of the final target web service as the link text, resulting in a link mismatch. Thus, it cannot be expected that email clients block such emails. Yet, they could take the misleading elements out or inform users about their presence , e.g., by means of a warning dialog that actively asks the user whether they want to continue.

Fake Tooltip, Link Mismatch, HTML Base-Tag, HTML Form-Tag. For these deception techniques, the email client would notice that the target URL and the URL provided do not match and could react. This is done, for example, by comparing the link hyperreference (Link Mismatch) attribute, the full URL (HTML Base-Tag), or the form action attribute (HTML Form-Tag) with the link text or the link title attribute (Fake Tooltip). Fake tooltips created using CSS (instead of the title attribute) can be completely deactivated by restricting the CSS properties or selectors (e.g., the hover selector).

Exceedingly Long, Double File-Extension. In case of very long URLs and attachment names (see Section 3.3.23) or misleading attachment names (see Section 3.3.14), the relevant parts might not be

[4] N-level-domains can have up to 63 characters; the whole domain section (all N-level-domains separated by ".") can have up to 256 characters overall.

[5] To the best of our knowledge, there is no restriction in the length of file names.

displayed. Email clients can prevent this by always displaying these parts, e.g., by omitting parts of the domain or attachment name.

Homographic Spoofing, Right-to-Left-Override, Hexadecimal Encoding. Inconsistencies, such as mixing characters from different alphabets in one word, can also be detected. In such cases, only characters from one alphabet can be displayed. This includes the deception technique Homographic Spoofing (see Section 3.3.16), as well as right-to-left overriding (see Section 3.3.17). The latter only makes sense in alphabets in which letters are written and read from right to left, such as in Arabic languages, but not in those in which Cyrillic or Latin letters are used. In case of hexadecimal encoding parts of a domain (see Section 3.3.4), an email client can inform the user about the inconsistency or simply decode the hexadecimal encoding.

Ropemaker. Remote resources changing the email content can be avoided by disabling loading remote content by default and asking the user if remote content should be loaded when necessary.

Mailto-Scheme. An email client can inform the user about adding attachments when preparing an email based on mailto links or even disable this feature altogether.

URL Userinfo Field. The email client can detect that the user info field is used. It can then either not display the user info or inform the user about potentially suspicious elements.

Sender-Name Spoofing. The sender name is not sufficient to determine the sender, as it can be spoofed even with SPF. The email client should display both, the sender's name and the email address.

5. Analysis of email client susceptibility

Ideally, all of the deception techniques that can be addressed by email clients, should be addressed by today's email clients. In this section, we present an analysis of the susceptibility of several email clients to the 13 deception techniques considered relevant in the previous section (Section 4.2).

5.1. Methodology

5.1.1. Preparation of the EML files

We chose a manual inspection of a representative sample of email clients as opposed to a large sample, since we wanted to inspect each client in detail. To prepare the analysis, we implemented instantiations for each of the 13 deception techniques considered relevant in the previous section (Section 4.2) as EML files[6] and then uploaded these EML files into a GMail account. We uploaded the files using the Internet Message Access Protocol (IMAP) in Mozilla Thunderbird by dragging the EML files directly into the inbox on a desktop computer. We never send an email from one email account to another. Note that there is more than one type of implementation for some deception techniques. Therefore, we implemented

- Homographic Spoofing and Mailto-Scheme using UTF encoding and Punycode encoding
- Fake Tooltip once using the HTML "title" attribute and once using CSS
- Mailto-Scheme is implemented once using "attach" parameter and once with "attachment" both mentioned in Müller et al. (2020).

5.1.2. Sample of clients

We selected email clients to gain a sample with a broad representation of the market. First, we included those email clients with the highest market share (together close to 90%[7]), i.e., Microsoft Outlook, Gmail, and Apple Mail. We complemented these with lesser used email clients that represent specific niches in the market that we deemed of interest, i.e., Canaray (specifically marketed as security-focused), Mozilla Thunderbird (the open source client with the highest market share), and Mutt (a text-based client). Below, we provide a more detailed reasoning for each of the clients:

- *Apple Mail* for *iOS* and *macOS*. as it is email client with the highest market share of 65.1% as of November 2022 when we conducted the testing and 50.0% at the time of writing.[7] Note that we include the desktop and the mobile version, since privacy measures by Apple make it hard to distinguish between both for the providers of email market share statistics and therefore statistics are typically for both clients together. We used the version of Apple Mail that comes with iOS 16.1.1 and Version 14.0 (3654.120.0.1.13) on macOS.
- *GMail* as *webmailer* which represents the second most frequently used email client with a market share of 23.5% as of November 2022 when we conducted the testing and 33.4% at the time of writing.[7] To aid comparison to the mobile platform, we also included the mobile app. We used the web version of mail.google.com from November 2022 (accessed using Mozilla Firefox) and as mobile app GMail 6.0.221016 on iOS 16.1.1.
- *Microsoft Outlook* for Windows as it has the highest market share for local email clients on Windows (3.4% as of November 2022 when we conducted the testing and 4.5% at the time of writing[7]). We used Microsoft Outlook Version 2209 (Build 16.0.15629.20256) on Windows 11.
- *Mozilla Thunderbird* as the open source email client with the highest market share (< 0.1% as of November 2022 when we conducted the testing and 0.2% at the time of writing[7]) and a popular choice in the Open source Community (Mayer et al., 2022). We used Mozilla Thunderbird 102.4.2 on Windows 11.
- *Canary mobile app* as the only commonly used iOS App with GPG support and is advertised as secure email client. We used Canary 3.73 on iOS 16.1.1.
- *Mutt* console client as according to a survey among tech-savvy users we know that this client is in relatively wide use there. We used Version Mutt 2.2.6 installed using apt on Ubuntu 22.10 in the Gnome Terminal 3.42.2.

5.1.3. Preparation of the email clients

Apart from the Gmail webmail client, all other email clients in our analysis were configured to sync emails from the Gmail account. For the Gmail web interface, this step was unnecessary, as it automatically displays the inbox emails.

We installed the email clients and the Firefox browser (for accessing Gmail webmail) using the installers provided on their respective websites, and for mobile clients and macOS, from the Google Play Store and Apple App Store. Mutt was installed using *Debian's Advanced Packaging Tool (apt-get)*. Aside from configuring the email sync, all installations were performed using the default settings without further modifications.

The mobile devices (*Samsung S20 SM-G981B/DS* and *Apple iPhone SE MX9U2ZD/A*), along with the macOS system (*Mac Mini A1347*) and Ubuntu Linux, are physical test devices, all updated to the latest software versions. Microsoft Windows was freshly installed in a virtual machine using Microsoft Hyper-V Manager (Version 10.0.22621.1).

[6] The used EML files are uploaded and can be opened by a regular text editor or email clients: https://github.com/SecUSo/paper-artifacts-email-deception-techniques/tree/main/2.Analysis/1.Source-EML-Files.

[7] The State of Email Client Market Share, https://www.litmus.com/email-client-market-share.

5.1.4. Analysis procedure

For the analysis, emails were not only displayed but also interacted with when possible. The steps performed were consistent across all email clients, with variations only in the specific interactions. It is important to note that we did not try to simulate user behavior in the interactions we performed. Instead the interactions aimed at discovering the available UI elements, such as tool tips or warning banners that appeared. We discuss this aspect further in Section 6.4.

Each email representing a deception technique (initially provided as an EML file) was opened sequentially. Screenshots were captured first upon opening the email and again after any interaction that altered the display. Interactions varied depending on whether the deception technique involved the sender, links, or attachments, and whether a mobile or desktop client was used.

For sender-based deception techniques, there were no links or attachments in the EML file, so interactions were limited to sender field. In link-based deception techniques, only the links were interacted with, ignoring the sender and attachments. Similarly, in attachment-based deception techniques, only the attachments were interacted with, ignoring the sender and links.

Interactions also differed between desktop systems (including Apple Mac and the webmailer client) and mobile devices due to their input methods. On desktop systems, security indicators (such as sender information, links, and attachments) could be hovered over (by moving the mouse cursor over them) and clicked (by pressing the left mouse button while the mouse cursor is over them). On mobile devices, these same security indicators could be interacted with via a long tap (holding a finger on the screen at the position of the security indicator for about two seconds) or a regular tap (briefly touching the screen at that position).

If a warning was displayed during any interaction, a screenshot was taken, and the highlighted option – or, if none was highlighted, the first option (typically the one higher or further to the left) – was tapped or clicked.

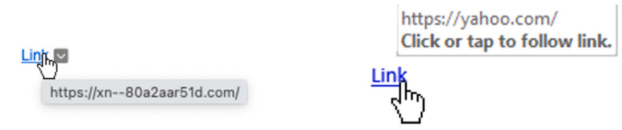
In summary, each email was opened, and the senders, links, and attachments were either hovered over and clicked on desktop systems or tapped and long-tapped on mobile devices.

The screenshots captured during this process were sorted by deception technique and email client to be used for labeling in the next step.[8]

5.1.5. Labeling procedure

For labeling a spreadsheet[9] was created and each deception technique on each email client was listed in one row. Beside the email client name, deception technique, path to EML file (Section 5.1.1) and screenshots taken (Section 5.1.4) the columns for labeling and a calculated score for the susceptibility rating (Section 5 are used. The labeling (Section 5.2 was done by opening the respective screenshots of the email client and deception technique in a regular image viewer under windows by the authors. It was solely labeled by the information shown in the screenshot in connection with the interaction done without further interpretation to ensure a high level of objectivity. The labeling based on the categorization and their aspects and their expressions is described in the following section in detail.

To facilitate labeling, a spreadsheet was created, listing each deception technique for every email client in individual rows. This spreadsheet includes columns for the email client name, deception technique, path to the EML file (described in Section 5.1.1), screenshots taken (described in Section 5.1.4), labeling (described in Section 5.2), and a calculated susceptibility score (described in Section 5).



Link

https://xn--80a2aar51d.com/

https://yahoo.com/
Click or tap to follow link.

Link

(a) correct information (AppleMail on macOS)

(b) misleading information (Microsoft Outlook)

Figure 11. In case (a), only the correct information (encoded in Punycode) is displayed, whereas in case (b), only the misleading information (yahoo.com in Cyrillic letters) is visible to the user.

The labeling process involved opening the corresponding screenshots of the email client and deception technique in the standard image viewer on Microsoft Windows. The authors labeled the screenshots based solely on the information displayed, in connection with the interactions performed, to maintain a high level of objectivity without further interpretation.

The criteria for labeling, including the categorization and its aspects, will be described in detail in the following section.

5.2. Results – Susceptibility categorization of deception techniques

Our analysis identified three key aspects that systematically categorize the susceptibility of email clients with respect to information about the sender, links, and attachments: (a) the accessibility of correct information to the user, (b) the amount of user interaction required to discover the correct information (particularly when more effort is needed to access correct information compared to incorrect information), and (c) the level of support provided by the email client. In the following sections, we will describe each of these aspects in detail, along with their possible expressions. Fig. 19 gives an overview of the analysis results.

5.2.1. Accessible information

This aspect indicates which information is accessible to the user. There can be both correct and misleading information, or neither may be present. This results in four possible expressions: (1) only the correct information is displayed to the user; (2) only the misleading information is displayed to the user; (3) both, the correct and misleading information are displayed to the user; and (4) neither the correct nor the misleading information are displayed to the user. Our analysis has shown examples for all of these four expressions.

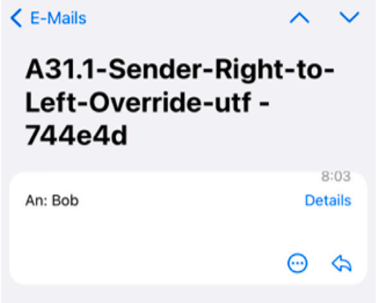
A prominent example where only the correct information is shown is the use of Homographic Spoofing (UTF) in links in Apple Mail on macOS. An example is shown in Fig. 11(a), where the user is not mislead by a domain similar looking to yahoo.com in the tooltip of Apple Mail (in macOS), but instead the punycode encoded correct domain is shown. The correct information is accessible here right after hovering the link with the cursor. By hovering the link, the URL with the punycode encoded domain is displayed.

Similarly, Mozilla Thunderbird shows only the misleading information. By hovering the link, the URL with the domain in Homographic Spoofing (Punycode) is displayed as looking like *yahoo.com* but it is in fact written using Cyrillic letters. An example is shown in Fig. 11(b), where the user is mislead by a domain similar looking to yahoo.com in the statusbar of Microsoft Outlook. Only the misleading information is accessible here. For these two cases, the former one is ideal, because there is no possibility of the user being misled. In contrast, the latter represents a strong susceptibility of the client to the deception technique.

The remaining two cases are more nuanced. Showing no information at all (i.e., neither the correct nor the misleading information)

[8] Screenshots of interactions published: https://github.com/SecUSo/paper-artifacts-email-deception-techniques/tree/main/2.Analysis/2.Evaluation-Screenshots.

[9] The spreadsheet for labeling the screenshots: https://github.com/SecUSo/paper-artifacts-email-deception-techniques/blob/main/2.Analysis/3.FindingsLabeling.xlsx.



E-Mails

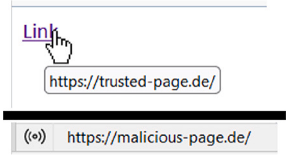
A31.1-Sender-Right-to-Left-Override-utf - 744e4d

8:03

An: Bob

Details

Figure 12. Example where neither the correct nor the misleading information is shown for Right-to-Left-Override (UTF) evaluated in Canary.



Link

https://trusted-page.de/

https://malicious-page.de/

Figure 13. Example where both correct information and misleading information is shown at the same level of interaction for Link Mismatch in Mozilla Thundebird.

could confuse the user. For example, when using the Right-to-Left-Override character in the sender email, Canary shows no sender at all. This is shown in Fig. 12. Here, neither the correct nor the misleading information is accessible.

Similarly, when both the correct and misleading information are available, this can lead to confusing situations, particularly if seeing the misleading information requires less interaction than the correct information (cf. Section 5.2.2). Fig. 13 shows an example where both the correct and misleading information are available at the same level of interaction.

5.2.2. *Interaction required*

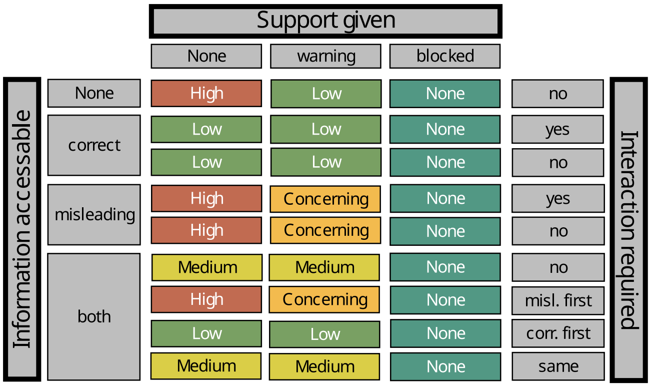
This aspect indicates if any user interaction is needed to discover the correct information. It addresses not only the availability of the correct information but also how difficult it is to access. The level of interaction needed can be described as a Boolean value for both correct and misleading information (if accessible). However, differences in email client User Interfaces make detailed quantification challenging and open to interpretation. If both the correct information and misleading information are accessible, four expressions are possible: (1) no interaction is required to display either piece of information, (2) the misleading information is displayed with less interaction than the correct information, (3) the correct information is displayed with less interaction than the misleading information, or (4) both are displayed with the same level of interaction.

5.2.3. *Support given*

This aspect indicates which support (if any) is offered by the email client. During the analysis three expressions emerged here: (1) no support; (2) a warning message is displayed to the user relating to one or multiple elements in the email (i.e., sender, link, or attachment) but allows interaction with the respective content; and (3) the email client blocks the interaction with one or multiple elements in the email.

5.3. *Results – Security rating of susceptibility*

When combined, the three aspects and their corresponding expressions described in the previous section allow a security rating assessment of the deception techniques in each email client.[10] The following security rating emerge (See also visualization in Fig. 14).



Information accessible	Support given: None	Support given: warning	Support given: blocked	Interaction required
None	High	Low	None	no
correct	Low	Low	None	yes
	Low	Low	None	no
misleading	High	Concerning	None	yes
	High	Concerning	None	no
both	Medium	Medium	None	no
	High	Concerning	None	misl. first
	Low	Low	None	corr. first
	Medium	Medium	None	same

Figure 14. Proposed rating of susceptibility.

Figure 15. Example where the email client disables the link so it is non-clickable for HTML Form-Tag evaluated in Microsoft Outlook.

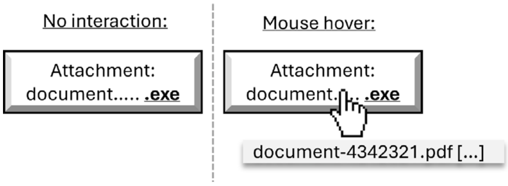

No interaction:
Attachment: document..... .exe
Mouse hover:
Attachment: document..... .exe
document-4342321.pdf [...]

Figure 16. Example where both the correct information and the misleading information is shown, but the misleading information is shown only after hovering. This did not occur in any email client of our evaluation.

5.3.1. *None*

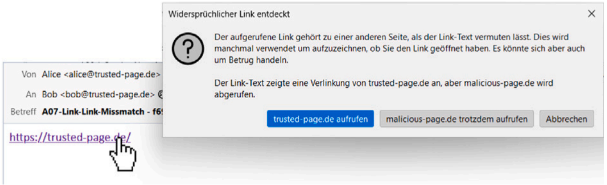
This level encompasses cases in which the *email client blocks deception techniques completely* (like for HTML Form-Tag in Microsoft Outlook in Fig. 15). The defining characteristic of this level is that there is no potential for incorrect decisions by the user. While this is the most desirable case from a security perspective, overzealous blocking can lead to false positives which could impact the usability or utility of the respective email client. In Fig. 19 cases with these levels are marked in green.

5.3.2. *Low*

This level comprises three types of cases: (1) cases where either *only the correct information is accessible* by the user (Fig. 11(a)); (2) both the correct and misleading information are accessible to the user, but *the correct information is accessible with less interaction required* (Fig. 16); or (3) no information is provided, but there is a warning message.

If the correct information is presented, the user can make a informed decision, although adequate security awareness training may still be necessary. Furthermore, we contend that users typically cease interaction once they have obtained the necessary information, thereby decreasing the likelihood of encountering misleading information that requires further engagement. In cases where no information is provided but a warning is issued, users can still make an informed decision based on the warning alone. This is why such cases are also included in this category. In Fig. 19 cases with this level are marked in light green.

[10] Automatic calculation of security rating using excel sheet - https://github.com/SecUSo/paper-artifacts-email-deception-techniques/blob/main/2.Analysis/3.FindingsLabeling.xlsx.



Widersprüchlicher Link entdeckt

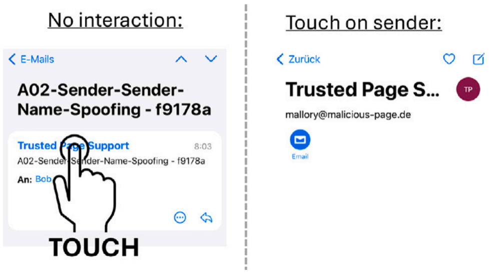
Der aufgerufene Link gehört zu einer anderen Seite, als der Link-Text vermuten lässt. Dies wird manchmal verwendet um aufzuzeichnen, ob Sie den Link geöffnet haben. Es könnte sich aber auch um Betrug handeln.

Der Link-Text zeigte eine Verlinkung von trusted-page.de an, aber malicious-page.de wird abgerufen.

Von Alice <alice@trusted-page.de>
An Bob <bob@trusted-page.de>
Betreff **A07-Link-Link-Missmatch - f6**

https://trusted-page.de/

Figure 17. Example where the misleading information is shown, but a warning is displayed to the user. Shown here, the Link Mismatch deception technique.



A02-Sender-Sender-Name-Spoofing - f9178a

Trusted Page Support 8:03
A02-Sender-Sender-Name-Spoofing - f9178a
An: Bob

Trusted Page S...
mallory@malicious-page.de

Figure 18. Example where both the correct information and the misleading information is shown, but the correct information is only shown after a tap on the misleading information, revealing the actual sender in this case of Sender-Name Spoofing in Canary.

5.3.3. Medium

This level is assigned when *both the correct and misleading information are accessible to the user at the same level of interaction* (i.e., both immediately visible or both appear in the same dialog). Fig. 13 shows an example of this. In Fig. 19 cases with this level are marked in yellow.

5.3.4. Concerning

This level includes cases where a warning is displayed after the misleading information is presented, as illustrated in Fig. 17. The warning provides the user with additional information, helping them make a more informed decision rather than relying solely on the misleading information. In Fig. 19 cases with this level are marked in dark orange.

5.3.5. High

This level includes cases where the user either sees only the misleading information (see Fig. 11(b)) or no information at all, as in Fig. 12.

As discussed in Section 5.3.2, we argue that users typically stop interacting once they believe they have obtained the necessary information, reducing the likelihood of uncovering correct information that requires further action. An example is shown in Fig. 18

To illustrate the security implications of no information being displayed, consider a link where the target URL is hidden, and the user is unaware of this. The user may overlook the absence of this information, such as in the status bar, potentially leading to a higher risk of deception.

In Fig. 19 cases with this level are marked in red.

5.4. Results – detailed susceptibility

Fig. 19 shows the results regarding the email clients' susceptibility to the deception techniques. Each case (cell in the table) is colored according to the security rating as described in the previous section. There are 67 high cases (red) out of the 176 cases. Only 70 were none or low (green). The remaining 39 were in medium (yellow).

Overall, deception techniques based on the link and sender seem to be far more problematic than those based on attachments, with 41.3% and 43.8% of cases being high respective compared to the 12.5% for the attachments. Yet, this does not paint the complete picture, since the deception techniques based on links exhibit the highest ratio of none cases with 22.1% and the deception techniques based on the sender exhibit the highest ratio of low cases with 39.6%. The skew in the none cases of the link-based deception techniques are mostly due to the *Mailto-Scheme* deception technique. For the sender the *Right-to-Left-Override* deception technique has a similar, albeit smaller effect on the skew of the overall numbers of the sender-based cases.

Arguably, the most problematic of the deception techniques we tested is the *Homographic Spoofing* deception technique with the UTF encoding. It is high in all but one of the evaluated email clients. This also includes Mutt which in all other cases simply displays the source code of the email, however in this case the email domain is encoded in punycode in the source code and Mutt displays the decoded version which allows human indistinguishable characters.

Comparing the fake tooltip deception technique for the two email clients tested under Windows, the following edge case emerges. In the first email client (Microsoft Outlook in Microsoft Windows), only the misleading information is present before interaction, while in the second email client (Mozilla Thunderbird) it is present only together with the correct information after the interaction, resulting – strictly following the categorization 5.2 – in two different levels of the security rating, concerning and medium.

In contrast to the most problematic deception techniques, the additional parameter for the attachment in the Mailto-Scheme deception technique is blocked by the two email clients tested under Windows, Mutt on Ubuntu and Apple Mail on macOS. On the webmailers and the mobile clients this deception technique cannot work because they have either no access to the local file system or do not provide it to the email client without user interaction. Therefore, all cases relating to this deception technique are rated as none. This is only one example of a frequently emerging theme we found during the evaluation: platform plays a role regarding the susceptibility of the deception techniques. In this case, the permission system on the mobile platforms prevents this deception technique.

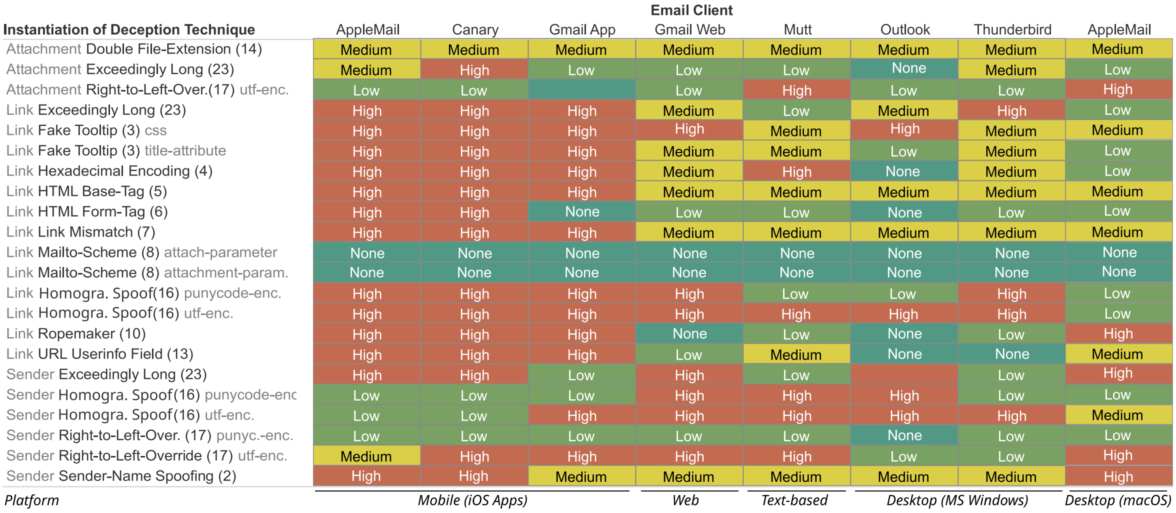
Similarly, deception techniques using links are either none or high for all mobile clients. The platform has a large impact, e.g., iOS mobile clients by default access the website already on long press (link preview) which explains their high susceptibility.

Another aspect in which the platform plays a role is that all tested mobile email clients fail to display the sender email address without interaction, while all desktop email clients and the GMail webmailer directly show the email address and the sender name.

These platform-dependent results are also illustrated by the mobile clients Canary and Apple Mail having difficulties showing long information, even after an interaction. While one might argue that this is due to the smaller screen, the GMail mobile app proves this wrong by showing the whole long sender address after expanding the email header section. In contrast, Outlook and the GMail webmailer do not show the complete sender if it is too long. Gmail webmail does not even after the interaction. However, exceedingly long links are shown completely or at least the registered domain name.

The GMail mobile app and Outlook block the *Link-HTML-Form-Tag* completely while Thunderbird and the GMail webmailer show a warning. Canary as well as Apple Mail on iOS are vulnerable to this deception technique even if the preview is disabled, meaning that they do not show where the link refers to. A particularly problematic behavior to this deception technique is shown in the Canary email client, which loads the webpage of the target URL and displays it completely in the email window (including JavaScript execution).

A few deception techniques, such as the *Fake-Tooltip*, aim at exploiting CSS. All clients tested under Windows, MacOS and the GMail webmail client allow using CSS to show an artificial tooltip with a spoofed URL beside the real URL (see Fig. 20). Outlook only shows the fake tooltip using CSS (without hovering) but blocks the one created using the HTML title attribute. Apple Mail on MacOS displays the fake



Instantiation of Deception Technique	AppleMail	Canary	Gmail App	Email Client Gmail Web	Mutt	Outlook	Thunderbird	AppleMail
Attachment Double File-Extension (14)	Medium	Medium	Medium	Medium	Medium	Medium	Medium	Medium
Attachment Exceedingly Long (23)	Medium	High	Low	Low	Low	None	Medium	Low
Attachment Right-to-Left-Over.(17) utf-enc.	Low	Low		Low	High	Low	Low	High
Link Exceedingly Long (23)	High	High	High	Medium	Low	Medium	High	Low
Link Fake Tooltip (3) css	High	High	High	High	Medium	High	Medium	Medium
Link Fake Tooltip (3) title-attribute	High	High	High	Medium	Medium	Low	Medium	Low
Link Hexadecimal Encoding (4)	High	High	High	Medium	High	None	Medium	Low
Link HTML Base-Tag (5)	High	High	High	Medium	Medium	Medium	Medium	Medium
Link HTML Form-Tag (6)	High	High	None	Low	Low	None	Low	Low
Link Link Mismatch (7)	High	High	High	Medium	Medium	Medium	Medium	Medium
Link Mailto-Scheme (8) attach-parameter	None	None	None	None	None	None	None	None
Link Mailto-Scheme (8) attachment-param.	None	None	None	None	None	None	None	None
Link Homogra. Spoof(16) punycode-enc.	High	High	High	High	Low	Low	High	Low
Link Homogra. Spoof(16) utf-enc.	High	High	High	High	High	High	High	Low
Link Ropemaker (10)	High	High	High	None	Low	None	Low	High
Link URL Userinfo Field (13)	High	High	High	Low	Medium	None	None	Medium
Sender Exceedingly Long (23)	High	High	Low	High	Low		Low	High
Sender Homogra. Spoof(16) punycode-enc	Low	Low	Low	High	High	High	Low	Low
Sender Homogra. Spoof(16) utf-enc.	Low	Low	High	High	High	High	High	Medium
Sender Right-to-Left-Over. (17) punyc.-enc.	Low	Low	Low	Low	Low	None	Low	Low
Sender Right-to-Left-Override (17) utf-enc.	Medium	High	High	High	High	Low	Low	High
Sender Sender-Name Spoofing (2)	High	High	Medium	Medium	Medium	Medium	Medium	High
Platform	*Mobile (iOS Apps)*			*Web*	*Text-based*	*Desktop (MS Windows)*		*Desktop (macOS)*

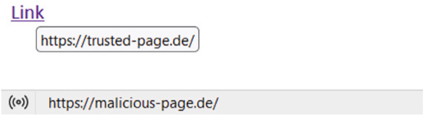
Figure 19. Security rating of deception techniques by email client from none to high (Link to detailed findings).

tooltip slightly before the real one when hovering, which may cause users to stop hovering prematurely, seeing only the fake tooltip with a seemingly trustworthy URL while missing the real tooltip that reveals the actual malicious URL.

Another deception technique based on CSS is *Ropemaker*. However, in this case, CSS is loaded as an external resource, so for the deception technique to work the client has to allow loading remote content. Besides the security risk, this already represents a privacy leak. Additionally, Apple Mail and Canary loaded the remote content and allowed to change the email (i.e., the link) based on the remote content.

When looking at whether email clients offer warnings, a few findings are of note. Firstly, only Thunderbird offers assistance in preventing *Link Mismatch* deception techniques. It shows an active warning and offers the user to access the website referred to by the link text rather than the real target URL. Secondly, no evaluated email client assists the user in reading the URL by highlighting the domain as it is common for, e.g., the address bar of many web browsers. Thirdly, some email clients had mechanisms to warn against or block the *URL Userinfo Field* deception technique. Thunderbird removed the userinfo from the link to prevent the deception technique. Outlook is even more strict and disables the link if there is a userinfo. The GMail webmailer shows an active warning when the link is clicked. Unfortunately, this warning does not carry over to the GMail mobile iOS app. Fourthly and on a more positive note, four of the seven email clients support the user in checking the file extension with an icon. A notable exception is Canary, which fails to show the last part in the *Exceedingly Long* deception technique for the attachment and also does not show a meaningful icon with the file extension. Another notable case is Mutt. It does not support users with icons based on the file extension. However, it relies rather on the MIME Content-Type of the attachment rather than the file extensions when opening attachments. User configuration may be required to open attachments in anything other than a text editor.[11]

One specific issue that can arise in the context of internationalization represents the *Right-to-Left-Override* deception technique. The GMail mobile app does not show the sender field at all when the respective control character encoded as UTF-8 is included there. Thus, this email client is rated as high. The GMail webmailer and also Apple Mail support the control characters and show the email address in reverse order which allows spoofing an email even if secured with SPF. The possibility to circumvent these protocols is particularly problematic and the respective clients are therefore rated as high in this regard. In contrast, the *Right-to-Left-Override* deception technique did not work at all for the attachment on most email clients and also most of the email clients displayed a warning. The GMail app blocked opening the attachment entirely. Mutt does not show the file name including the file extension, but it shows the mime type.



Link

https://trusted-page.de/

https://malicious-page.de/

Figure 20. Thunderbird shows a fake tooltip and the real URL.

6. Discussion

We first summarize the answers to our research questions, then discuss further implications of our findings, as well as the limitations of our work.

6.1. Answering the research questions

RQ1. *Which deception techniques have been reported in the literature of the past decade that aim at deceiving users about the identity of senders, the targets of links, or the nature of attachments?*

To answer RQ 1, we conducted a systematic literature review and cataloged the deception techniques in emails identified in the literature of the past decade. Overall, we found 23 different deception techniques, showing a wide variety of ways to deceive users regarding the identity of senders, the targets of links, and the nature of attachments.

RQ2. *Are common email clients susceptible to these deception techniques?*

Answering this research question comprised multiple steps. First, we classified the deception techniques into those for which email clients can offer support to the users and those for which this is not feasible. Our analysis indicates that email clients can offer support for 13 of the deception techniques identified in the literature review.

We then investigated the susceptibility of a sample of email clients to the 13 deception techniques for which email clients can offer support, and found that all but 2 of the deception techniques are still viable. Based on the security rating proposed in Section 5.3, we found that 38.1% of cases appeared to have a high susceptibility. Moreover, we found that deception techniques based on links were overall the most problematic, with 41.3% of the link-based deception technique

[11] This is specific to each Linux distribution, e.g., in Ubuntu that we use for our testing, the pdf viewer evince installs a respective configuration to open pdf attachments.

cases being rated as high. We saw in our analysis that for several deception techniques the platform (i.e., mobile email client app vs. webmail client vs. desktop email client) also plays a big role. Email client apps were much more susceptible to link-based attacks and account for a lot of the high susceptibility cases of these deception techniques. This can be attributed to the lack of an equivalent to hovering on links to get more information, an interaction very common on desktop platforms. Earlier versions of iOS and Android showed the full target URL of a link when the user long-pressed it. In today's versions, this has been replaced by a preview (which can be disabled, though). While this is a convenience feature, we argue that such a preview can have negative security implications. Firstly, the current implementation on iOS only shows the registered domain of the target URL without an option to show the full URL, making it even harder to detect the *Redirect* deception technique due to the lack of further information usually present in the path of redirect URLs (e.g. /?url=targeturl). Secondly, a well-crafted imitation of a legitimate webpage familiar to a user might visually take precedence over the target URL, luring them onto dangerous websites.

6.2. The role of users of email clients

Our investigation has uncovered that 10 of the deception techniques identified in our systematic literature review cannot be addressed in the email client. Of these 10 deception techniques, only the *Sender-Mail Spoofing* deception technique can be addressed by the server by using SPF. However, the other deception techniques in this category either rely on ground truths that are usually not available (e.g., having a complete list of registered domains of all benign web services) or require external context (e.g., if a certain file type is expected as an attachment in a specific email). This indicates that a lot of the responsibility to counteract the deception techniques is offloaded onto the user. We argue that minimizing users' burden and the effort required to perform appropriate checks should be the responsibility of the email client (or server) whenever possible. If, instead, the effort piles up on the users, they might be overwhelmed, which would lead to negative consequences (Herley, 2009). Unfortunately, email clients do not have access neither to these ground truths nor the external context. Thus, users have to decide what to do on their own (e.g., whether to click a link or not). In order to enable users to make these decisions, effective security awareness and training measures that offer actionable strategies for these deception techniques are required. Potentially, such information could be integrated into email clients to offer at least some support to users by giving them instructions on what to check if they are unsure. Our list of the 10 deception techniques that cannot be covered by the email clients can guide developers of awareness and training materials towards creating materials that can enable users to defend against these deception techniques until (hopefully) technological progress enables solutions that do not require user interaction.

6.3. Role of developers of email clients

The developers are the fundamental part of defending against the 13 deception techniques that allow client support without additional information. However, we argue that developers need support in this regard as well (Green and Smith, 2016). In a first effort to support the developers in improving the situation, we disclosed our findings and provided them the EML files of our deception techniques instantiations of the email clients used in our analysis. We also gave them the usual 90 days period.[12] to fix the issues before publishing our findings. This period is over at the time of this submission. For Mutt, a solution has already been implemented for the deception technique Homographic Spoofing In this case, the configuration was adjusted so that Non-ASCII Character URLs are encoded in Punycode by default. This process took less than 24 h and serves as an example of how efficiently such adjustments can be made by assisting developers in finding a solution.

Defending against other deception techniques might need a more general discussion. For instance, a recurring theme in our analysis is an increased susceptibility to deception techniques through email client features that either (1) go beyond the specifications or (2) are arguably not strictly necessary to the email client functionality. While the *Mailto-Scheme* deception technique is the prime example of (1), the best illustration of (2) is loading remote content that could be easily integrated into the email itself. The latter not only has security implications by enabling attacks such as Ropemaker, but also impact the users' privacy, by allowing user tracking. Another example of such "features" is supporting CSS markup like `display: none` or `visibility: hidden`, which arguably have no legitimate use in emails. Conforming to specifications in the email domain is difficult enough without going beyond them with features. Instead, the example of CSS markup shows that, in the case of email, restricting what is allowed to a reduced specification can have security and privacy benefits, without impacting legitimate use cases. Moving standards towards this is an important direction for the future.

6.4. Limitations

Like any literature review, our work cannot claim completeness of the findings. Including additional sources (e.g., search engines) and search terms might have yielded additional results. We were following best practices in the procedure of performing a systematic literature reviews (Kitchenham, 2004). However since the part of screening the literature is done manually by the authors this leaves an unavoidable remainder of subjectivity. We compensate for this by publishing our evaluation documentation[3] transparently, from the literature found to the results presented in Section 5.2, in order to facilitate reproducibility and traceability. We argue that by using the sources we have chosen, in combination with the search terms we have chosen, we find the most relevant and impactful deception techniques. We furthermore argue that the deception techniques we found are likely to remain relevant for a long time: the email ecosystem is unlikely to change substantially in the foreseeable future, so email clients will have to pick up the slack and implement measure to help users to defend themselves. This is also evidenced by the fact that emerging trends, such as the use of generative artificial intelligence for crafting more personalized phishing emails (Shibli et al., 2024), are largely based on Large Language Models exploiting the deception techniques we found in our systematic literature review. Furthermore, we argue that there seems to be a missing link between developers and researchers, as evidenced by the fact that some widely researched deception techniques are still applicable today in email clients. We tried to bridge this gap by confronting the developers of the email clients with our findings. Unfortunately, to the best of our knowledge, still most of the deception techniques found are not fully addressed in the latest updates.

The proposed security rating of susceptibility (Section 5.3) aims to make email clients and deception techniques comparable. However, like any security rating scale, it has inherent limitations. These include some loss of nuanced information and variability in individual user behavior — some users might recognize clues that others miss. That also lead to our decision to not try and simulate user behavior. Instead, our focus is on objective criteria, specifically the accessibility of the correct information needed by the user and the interaction required to obtain it. Conducting user studies to determine whether this information is effectively used to make correct decisions is a separate research question and falls outside the scope of this paper.

Similarly, our results have been influenced by the specific selection of email clients in our susceptibility analysis. Although including a larger sample of email clients – representing close to 90% of the market

[12] See, e.g., https://googleprojectzero.blogspot.com/p/vulnerability-disclosure-policy.html.

share, with greater diversity and coverage across more platforms – could have provided additional insights, the level of automation required to conduct such a comprehensive analysis was beyond the scope of this work.

Moreover, we evaluated the email clients in their default configuration, based on the premise that an email client should be secure by default. The question of how an email client enables users to make informed decisions about dynamic changes is a separate research issue. Both of these areas represent important avenues for future work to further validate our findings.

Some aspects of our analysis are arguably subjective in nature, since some of the cases in email clients must be considered fringe cases and could be interpreted in multiple ways. Yet, we make both the EML files of our analysis as well as the individual results pertaining to each email client in the form of screenshots publicly available in a GitHub repository. Thereby, we offer the greatest possible replicability and traceability of our findings.

Despite making the EML files available, replicability might still be hindered by the fact that the GMail webmail client and its email client app versions we used in our study might not be easily available at a later time. The only way to counter this effect would have been to not include these two platforms at all. However, we argue that the additional findings which are well documented by the publicly available screenshots outweigh this likely issue in replications.

7. Conclusion

In this work, we systematize and contextualize the current state of deception techniques in emails in several steps. We first present a systematic literature review of deception techniques in emails of the past decade, identifying overall 23 deception techniques in the literature. However, we found that only for 13 of these deception techniques email clients can offer support to users. Currently, the remaining 10 deception techniques require either the server or the user to take responsibility for defending against them. We tested the susceptibility of a representative sample of email clients to the 13 deception techniques for which email clients can offer support and found that the majority are still viable, leading to high ratings in 38.1% of the tested cases. Deception technique based on links seems to be the most problematic, with 41.3% of tested cases exhibiting a high rating. Additionally, we found that the platform used and “unnecessary” features presently play an important role in the susceptibility of email clients to the identified deception techniques. Modern browsers are aware of misleading URLs, try to avoid them, and improve how to display URLs (Lin et al., 2011; Hu et al., 2021b). Hu et al. showed that forming correct security policies for deceptive URLs is challenging but necessary (Hu et al., 2021b). In their user study, URLs that can bypass security policies were misleading to participants (Hu et al., 2021b). Our results show that modern email clients still have the potential to improve displaying URLs in emails. We suggest that email clients follow browsers when displaying URLs. On the other hand, our deception techniques and prototypes can be used to test security policies of browsers, too.

7.1. Security recommendations for developers

Our general security recommendations for developers are threefold:

1. **Displaying indicators that are difficult for the attacker to control:** File extensions of attachments and the main domain of a URL or a sender address are important to identify malicious content, and an attacker can only control them to a limited extent. File names, subdomains, and local parts are much easier to control. Therefore, file extensions and the main domain should always be recognizable and visible. Long file names and co can make this more difficult. The same applies to showing only the sender’s display name but not the sender’s address. Thunderbird, for example, shows display names and the sender’s address by default. Gmail always displays the file extension as an extra icon.
2. **Consideration of security recommendations:** Email is a collection of complicated RFCs with many standardized special cases. For example, the left-to-right representation or characters from different alphabets in the sender address or URL can be exploited. These special cases are important to make email accessible to the entire world population, not just the Western world. At the same time, they are used creatively to deceive users. Security considerations, such as Section 4 of RFC 6532, are briefly mentioned in RFCs and should be considered when implementing them. For example, AppleMail on iOS displays Punycode as the sender address. This can be particularly helpful if different alphabets occur in the same character string.
3. **Separation between information from the client and the sender:** HTML has tooltips and various other tags. The typical use case for tooltips is additional information when filling out forms. In the case of emails, however, these are rather unusual and can conflict with information provided by the client. For example, a tooltip controlled by the attacker was often displayed next to the link. For instance, an attacker-controlled tooltip may appear next to a link, posing a risk that users could mistake it for information provided by the client rather than the email itself. This issue is particularly problematic in Apple Mail on macOS, where the fake tooltip appears slightly before the legitimate one, potentially causing users to miss the real tooltip if they do not hover over the link long enough. To prevent misinterpretation, only HTML elements that cannot be easily confused with client-generated information should be displayed in emails.

These recommendations are very general, and specific recommendations for individual tricks can be found in Section 4.2. Clients’ specific implementations can be very different and depend on the design and platform. The positive examples in Fig. 19 are also recommendations for visual implementation, and we recommend considering them when developing the user interface. Finally, we would like to point out that there are open source extensions for some email clients (also known as add-ons) that have been shown to be effective in helping users avoid falling for deception techniques.[13] Developers can adopt the functionality and design of these open-source extensions to improve the email client.

We included our proof-of-concept emails to the replication package. Developers can use the raw emails (as EML-File) and the formal description (as JSON-File) to add them to their test suite. Therefore, email client distributor and developer can automatically test their new version and track changes between different versions. We informed the developers about our results and based on these test cases, developers can improve their specific visualization of file extensions, URLs and sender addresses.

7.2. Future research

Future research involves examining identified deception techniques on a larger sample of email clients to get a broader overview of the susceptibility of email clients currently available to users. One approach to accomplish this is to create an automation framework that emulates opening a set of EML files in pre-configured email clients and evaluate the outcome using *optical character recognition* (OCR) techniques. However, it will require considerable effort to establish the automated interactions in a reliable way for the framework. The community may enhance the framework to cover upcoming deception techniques and

[13] e.g. TORPEDO email client add-on. Evaluated in Volkamer et al. (2017) - https://github.com/SecUSo/torpedo.

novel email clients. Furthermore, a user study exploring the real-world impact of these deception techniques could assist in expanding and refining the initial categorization and assessment based on the security rating. A longitudinal observation of attacks can explore what deception techniques are used in practice and identify new upcoming deception techniques. Based on these results, new deception techniques can be described and added to the systematization. Our security rating of susceptibility can be extended with a rating for criticality in practice.

Our research is only a systematization of current knowledge about deception techniques and countermeasures. Researchers can adopt our attack description and formalize new attacks. In consequence, we can build a catalogue of known attacks. That can be used for user studies with phishing attacks, awareness material and developers to improve their applications.

In our paper, we proposed a Security Rating that evaluates the information displayed, the support provided by the email client, and the user interactions required. This rating is based on a structured argumentation process and can be refined in future research, such as through user studies. The rating is calculated using a spreadsheet, allowing for modifications to each combination of aspects and their corresponding expressions.[14]

CRediT authorship contribution statement

Maxime Fabian Veit: Writing – review & editing, Writing – original draft, Visualization, Validation, Supervision, Software, Resources, Project administration, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Oliver Wiese:** Writing – review & editing, Writing – original draft, Visualization, Validation, Supervision, Software, Resources, Project administration, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Fabian Lucas Ballreich:** Writing – review & editing, Writing – original draft, Visualization, Validation, Supervision, Software, Resources, Project administration, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Melanie Volkamer:** Writing – review & editing, Writing – original draft, Visualization, Validation, Supervision, Software, Resources, Project administration, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Douglas Engels:** Writing – review & editing, Writing – original draft, Visualization, Validation, Supervision, Software, Resources, Project administration, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Peter Mayer:** Writing – review & editing, Writing – original draft, Visualization, Validation, Supervision, Software, Resources, Project administration, Methodology, Investigation, Formal analysis, Data curation, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

[14] Spreadsheet with the tab ”SecurityRating“ for fine-tuning the rating in future research:https://github.com/SecUSo/paper-artifacts-email-deception-techniques/blob/main/1.LiteratureResearch/Literature_Combined_All_v0.6.2.xlsx.

References

Abedin, N.F., Bawm, R., Sarwar, T., Saifuddin, M., Rahman, M.A., Hossain, S., 2020. Phishing attack detection using machine learning classification techniques. In: 2020 3rd International Conference on Intelligent Sustainable Systems (ICISS). pp. 1125–1130. http://dx.doi.org/10.1109/ICISS49785.2020.9315895.

Agten, P., Joosen, W., Piessens, F., Nikiforakis, N., 2015. Seven months' worth of mistakes: A longitudinal study of typosquatting abuse. In: Proceedings 2015 Network and Distributed System Security Symposium. Internet Society, http://dx.doi.org/10.14722/ndss.2015.23058, https://www.ndss-symposium.org/ndss2015/ndss-2015.

Alazab, M., Broadhurst, R., 2016. Spam and criminal activity. In: Trends and Issues in Crime and Criminal Justice. vol. 526, pp. 1–20, ISSN 1836-2206.

Andryukhin, A., 2019. Phishing attacks and preventions in blockchain based projects. In: 2019 International Conference on Engineering Technologies and Computer Science (EnT). IEEE, pp. 15–19. http://dx.doi.org/10.1109/EnT.2019.00008, https://ieeexplore.ieee.org/document/8711898/.

Awasthi, A., Goel, N., 2021. Generating rules to detect phishing websites using URL features. In: 2021 1st Odisha International Conference on Electrical Power Engineering, Communication and Computing Technology(ODICON). pp. 1–9. http://dx.doi.org/10.1109/ODICON50556.2021.9429003.

Azeez, N., Ade, J., Misra, S., Adewumi, A., Vyver, C., Ahuja, R., 2020. Identifying phishing through web content and addressed bar-based features. Adv. Intell. Syst Comput 1016, 19–29.

Balan, G., Popescu, A.S., 2018. Detecting java compiled malware using machine learning techniques. In: 2018 20th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC). IEEE, pp. 435–439. http://dx.doi.org/10.1109/SYNASC.2018.00073, https://ieeexplore.ieee.org/document/8750755/.

Bhardwaj, A., Sapra, V., Kumar, A., Kumar, N., Arthi, S., 2020. Why is phishing still successful? Comput. Fraud Secur. 2020 (9), 15–19. http://dx.doi.org/10.1016/S1361-3723(20)30098-1.

Blum, A., Wardman, B., Solorio, T., Warner, G., 2010. Lexical feature based phishing URL detection using online learning. In: Proceedings of the 3rd ACM Workshop on Artificial Intelligence and Security AISec '10. Association for Computing Machinery, pp. 54–60. http://dx.doi.org/10.1145/1866423.1866434.

Chen, J., Paxson, V., Jiang, J., 2020. Composition kills: A case study of email sender authentication. pp. 2183–2199, https://www.usenix.org/conference/usenixsecurity20/presentation/chen-jianjun.

Clark, J., van Oorschot, P.C., Ruoti, S., Seamons, K., Zappala, D., 2021. SoK: Securing email—a stakeholder-based analysis. In: Borisov, N., Diaz, C. (Eds.), Financial Cryptography and Data Security. Springer Berlin Heidelberg, pp. 360–390.

Cross, C., Gillett, R., 2020. Exploiting trust for financial gain: An overview of business email compromise (bec) fraud. J. Financial Crime 27 (3), 871–884.

Du, Y., Xue, F., 2013. Research of the anti-phishing technology based on e-mail extraction and analysis. In: 2013 International Conference on Information Science and Cloud Computing Companion. pp. 60–65. http://dx.doi.org/10.1109/ISCC-C.2013.110.

Foster, I.D., Larson, J., Masich, M., Snoeren, A.C., Savage, S., Levchenko, K., 2015. Security by any other name: On the effectiveness of provider based email security. In: Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security - CCS '15. ACM Press, pp. 450–464. http://dx.doi.org/10.1145/2810103.2813607.

Franz, A., Zimmermann, V., Albrecht, G., Hartwig, K., Reuter, C., Benlian, A., Vogt, J., 2021. {Sok}: Still plenty of phish in the sea—a taxonomy of {user-oriented} phishing interventions and avenues for future research. In: Seventeenth Symposium on Usable Privacy and Security (SOUPS 2021). pp. 339–358.

Green, M., Smith, M., 2016. Developers are not the enemy!: The need for usable security apis. IEEE Secur. Privacy 14 (5), 40–46. http://dx.doi.org/10.1109/MSP.2016.111.

Group, A.-P.W., et al., 2005. Phishing activity trends report 4th quarter 2023. https://docs.apwg.org/reports/apwg_trends_report_q4_2023.pdf.

Gupta, N., Aggarwal, A., Kumaraguru, P., 2014. Bit.ly/malicious: Deep dive into short URL based e-crime detection. In: 2014 APWG Symposium on Electronic Crime Research (ECrime). pp. 14–24. http://dx.doi.org/10.1109/ECRIME.2014.6963161.

Hannay, P., Bolan, C., 2009. Assessment of internationalised domain name homograph attack mitigation. In: Australian Information Security Management Conference. http://dx.doi.org/10.4225/75/57b405aa30dee.

Heiderich, M., Frosch, T., Jensen, M., Holz, T., 2011. Crouching tiger - hidden payload: security risks of scalable vectors graphics. In: Proceedings of the 18th ACM Conference on Computer and Communications Security, CCS '11. Association for Computing Machinery, pp. 239–250. http://dx.doi.org/10.1145/2046707.2046735.

Herley, C., 2009. So long, and no thanks for the externalities. In: New Security Paradigms Workshop. pp. 133–144. http://dx.doi.org/10.1145/1719030.1719050.

Hu, H., Jan, S.T.K., Wang, Y., Wang, G., 2021a. Assessing browser-level defense against IDN-based phishing. pp. 3739–3756, https://www.usenix.org/conference/usenixsecurity21/presentation/hu-hang.

Hu, H., Jan, S.T., Wang, Y., Wang, G., 2021b. Assessing browser-level defense against {idn-based} phishing. In: 30th USENIX Security Symposium (USENIX Security 21). pp. 3739–3756.

Hu, H., Peng, P., Wang, G., 2018. Towards understanding the adoption of anti-spoofing protocols in email systems. In: 2018 IEEE Cybersecurity Development (SecDev). pp. 94–101. http://dx.doi.org/10.1109/SecDev.2018.00020.

Hu, H., Wang, G., 2018. End-to-end measurements of email spoofing attacks. pp. 1095–1112, https://www.usenix.org/conference/usenixsecurity18/presentation/hu.

Jakobsson, M., 2016. User trust assessment: a new approach to combat deception. In: Proceedings of the 6th Workshop on Socio-Technical Aspects in Security and Trust - STAST '16. ACM Press, pp. 73–78. http://dx.doi.org/10.1145/3046055.3046063.

Jampen, D., Gür, G., Sutter, T., Tellenbach, B., 2020a. Don't click: towards an effective anti-phishing training. a comparative literature review. Human-centric Comput. Inform Sci 10 (1), http://dx.doi.org/10.1186/s13673-020-00237-7.

Jampen, D., Gür, G., Sutter, T., Tellenbach, B., 2020b. Don't click: towards an effective anti-phishing training. A comparative literature review. Human-centric Comput. Inform. Sci 10 (1), http://dx.doi.org/10.1186/s13673-020-00237-7.

Khonji, M., Iraqi, Y., Jones, A., 2011. Lexical URL analysis for discriminating phishing and legitimate e-mail messages. In: 2011 International Conference for Internet Technology and Secured Transactions. pp. 422–427.

Khursheed, B., Pitropakis, N., McKeown, S., Lambrinoudakis, C., 2020. Microtargeting or microphishing? Phishing unveiled. In: Gritzalis, S., Weippl, E.R., Kotsis, G., Tjoa, A.M., Khalil, I. (Eds.), Trust, Privacy and Security in Digital Business. Springer International Publishing, pp. 89–105.

Kitchenham, B., 2004. Procedures for performing systematic reviews. Keele, UK, Keele University 33 (2004), 1–26.

Klensin, J., 2010. Rfc 5890: Internationalized domain names for applications (idna): Definitions and document framework.

Kokubun, Y., Nakamura, A., 2018. Analysis of malicious URLs on twitter. In: 2018 International Conference on Computational Science and Computational Intelligence (CSCI). IEEE, pp. 1285–1288. http://dx.doi.org/10.1109/CSCI46756.2018.00248, https://ieeexplore.ieee.org/document/8947647/.

Lee, J., Tang, F., Ye, P., Abbasi, F., Hay, P., Divakaran, D.M., 2021. D-fence: A flexible, efficient, and comprehensive phishing email detection system. In: 2021 IEEE European Symposium on Security and Privacy (EuroS & P). pp. 578–597. http://dx.doi.org/10.1109/EuroSP51992.2021.00045.

Li, X., Zhang, D., Wu, B., 2020. Detection method of phishing email based on persuasion principle. In: 2020 IEEE 4th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC), 1. pp. 571–574. http://dx.doi.org/10.1109/ITNEC48623.2020.9084766.

Lin, E., Greenberg, S., Trotter, E., Ma, D., Aycock, J., 2011. Does domain highlighting help people identify phishing sites? In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. pp. 2075–2084.

Marczak, W.R., Scott-Railton, J., Marquis-Boire, M., Paxson, V., 2014. When governments hack opponents: A look at actors and technology. In: 23rd USENIX Security Symposium (USENIX Security 14). USENIX Association, pp. 511–525, https://www.usenix.org/conference/usenixsecurity14/technical-sessions/presentation/marczak.

Maroofi, S., Korczyński, M., Hölzel, A., Duda, A., 2021. Adoption of email anti-spoofing schemes: A large scale analysis. IEEE Trans. Netw. Service Manag 18 (3), 3184–3196. http://dx.doi.org/10.1109/TNSM.2021.3065422.

Mayer, P., Poddebniak, D., Fischer, K., Brinkmann, M., Somorovsky, J., Sasse, A., Schinzel, S., Volkamer, M., 2022. I don't know why i check this." - investigating expert users' strategies to detect email signature spoofing attacks. In: Eighteenth Symposium on Usable Privacy and Security (SOUPS 2022). USENIX Association, Boston, MA, pp. 77–96, https://www.usenix.org/conference/soups2022/presentation/mayer.

Müller, J., Brinkmann, M., Poddebniak, D., Böck, H., Schinzel, S., Somorovsky, J., Schwenk, J., 2019a. Johnny, you are fired! – Spoofing OpenPGP and S/MIME signatures in emails. In: Proceedings of the 28th USENIX Conference on Security Symposium. USENIX Security Symposium. USENIX Association, pp. 1011–1028, https://www.usenix.org/system/files/sec19fall_muller_prepub.pdf.

Müller, J., Brinkmann, M., Poddebniak, D., Schinzel, S., Schwenk, J., 2019b. Re: What's Up Johnny?. In: Deng, R.H., Gauthier-Umaña, V., Ochoa, M., Yung, M. (Eds.), Applied Cryptography and Network Security. Springer International Publishing, pp. 24–42.

Müller, J., Brinkmann, M., Poddebniak, D., Schinzel, S., Schwenk, J., 2020. Mailto: Me your secrets. on bugs and features in email end-to-end encryption. In: 2020 IEEE Conference on Communications and Network Security (CNS). pp. 1–9. http://dx.doi.org/10.1109/CNS48642.2020.9162218.

Orman, H., 2012. Towards a semantics of phish. In: 2012 IEEE Symposium on Security and Privacy Workshops. pp. 91–96. http://dx.doi.org/10.1109/SPW.2012.12.

Pearson, E., Bethel, C.L., Jarosz, A.F., Berman, M.E., 2017. To click or not to click is the question: Fraudulent URL identification accuracy in a community sample. In: 2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC). pp. 659–664. http://dx.doi.org/10.1109/SMC.2017.8122682.

Pirocca, S., Allodi, L., Zannone, N., 2020. A toolkit for security awareness training against targeted phishing. In: Kanhere, S., Patil, V.T., Sural, S., Gaur, M.S. (Eds.), Information Systems Security. Springer International Publishing, pp. 137–159.

Pithawala, K., Jagtap, S., Cholachgud, P., 2021. Detecting phishing of short uniform resource locators using classification techniques. In: 2021 12th International Conference on Computing Communication and Networking Technologies (ICCCNT). pp. 1–5. http://dx.doi.org/10.1109/ICCCNT51525.2021.9579888.

Priya, S., Selvakumar, S., Velusamy, R.L., 2020. Gravitational search based feature selection for enhanced phishing websites detection. In: 2020 2nd International Conference on Innovative Mechanisms for Industry Applications (ICIMIA). pp. 453–458. http://dx.doi.org/10.1109/ICIMIA48430.2020.9074837.

Qabajeh, I., Thabtah, F., 2014. An experimental study for assessing email classification attributes using feature selection methods. In: 2014 3rd International Conference on Advanced Computer Science Applications and Technologies. pp. 125–132. http://dx.doi.org/10.1109/ACSAT.2014.29.

Reinheimer, B., Aldag, L., Mayer, P., Mossano, M., Duezguen, R., Lofthouse, B., v. Landesberger, T., Volkamer, M., 2020. An investigation of phishing awareness and education over time: When and how to best remind users. In: Sixteenth Symposium on Usable Privacy and Security (SOUPS 2020). pp. 259–284, https://www.usenix.org/conference/soups2020/presentation/reinheimer.

Salahdine, F., El Mrabet, Z., Kaabouch, N., 2021. Phishing attacks detection a machine learning-based approach. In: 2021 IEEE 12th Annual Ubiquitous Computing, Electronics Mobile Communication Conference (UEMCON). pp. 0250–0255. http://dx.doi.org/10.1109/UEMCON53757.2021.9666627.

Salloum, S., Gaber, T., Vadera, S., Shaalan, K., 2021. Phishing email detection using natural language processing techniques: A literature survey. Procedia Comput. Sci. 189, 19–28. http://dx.doi.org/10.1016/j.procs.2021.05.077, https://www.sciencedirect.com/science/article/pii/S1877050921011741.

Sangwan, S., Internet-wide study: State of spf, dkim, and dmarc (wave 6), https://redhuntlabs.com/blog/internet-wide-study-state-of-spf-dkim-and-dmarc/.

Sankhwar, S., Pandey, D., Khan, R.A., 2018. A novel anti-phishing effectiveness evaluator model. In: Satapathy, S.C., Joshi, A. (Eds.), Information and Communication Technology for Intelligent Systems (ICTIS 2017) - Volume 2. Springer International Publishing, pp. 610–618, https://link.springer.com/content/pdf/10.1007/978-3-319-63645-0.pdf.

Shen, K., Wang, C., Guo, M., Zheng, X., Lu, C., Liu, B., Zhao, Y., Hao, S., Duan, H., Pan, Q., Yang, M., 2021. Weak links in authentication chains: A large-scale analysis of email sender spoofing attacks. In: 30th USENIX Security Symposium (USENIX Security 21). pp. 3201–3217, https://www.usenix.org/conference/usenixsecurity21/presentation/shen-kaiwen.

Shibli, A.M., Pritom, M.M.A., Gupta, M., 2024. Abusegpt: Abuse of generative ai chatbots to create smishing campaigns. In: 2024 12th International Symposium on Digital Forensics and Security. ISDFS, pp. 1–6. http://dx.doi.org/10.1109/ISDFS60797.2024.10527300.

Shue, C.A., Kalafut, A.J., Gupta, M., 2008. Exploitable redirects on the web: identification, prevalence, and defense. In: Proceedings of the 2nd Conference on USENIX Workshop on Offensive Technologies. pp. 1–7.

Simoiu, C., Zand, A., Thomas, K., Bursztein, E., 2020. Who is targeted by email-based phishing and malware? measuring factors that differentiate risk. In: Proceedings of the ACM Internet Measurement Conference. Association for Computing Machinery, New York, NY, USA, pp. 567–576.

Singh, C., Meenu, 2020. Phishing website detection based on machine learning: A survey. In: 2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS). pp. 398–404. http://dx.doi.org/10.1109/ICACCS48705.2020.9074400.

Soni, P., Firake, S., Meshram, B.B., 2011. A phishing analysis of web based systems. In: Proceedings of the 2011 International Conference on Communication, Computing & Security, ICCCS '11. Association for Computing Machinery, pp. 527–530. http://dx.doi.org/10.1145/1947940.1948049.

Soussi, W., Korczyński, M., Maroofi, S., Duda, A., 2020. Feasibility of large-scale vulnerability notifications after GDPR. In: 2020 IEEE European Symposium on Security and Privacy Workshops (EuroS & PW). pp. 532–537. http://dx.doi.org/10.1109/EuroSPW51379.2020.00078.

Spaulding, J., Upadhyaya, S., Mohaisen, A., 2016. The landscape of domain name typosquatting: Techniques and countermeasures. In: 2016 11th International Conference on Availability, Reliability and Security (ARES). pp. 284–289. http://dx.doi.org/10.1109/ARES.2016.84.

Stransky, C., Wiese, O., Roth, V., Acar, Y., Fahl, S., 2022. 27 years and 81 million opportunities later: Investigating the use of email encryption for an entire university. In: To Appear in 43rd IEEE Symposium on Security & Privacy. SP'22, IEEE Computer Society, pp. 860–875.

Suriya, R., Saravanan, K., Thangavelu, A., 2009. An integrated approach to detect phishing mail attacks: A case study. In: Proceedings of the 2nd International Conference on Security of Information and Networks, SIN '09. Association for Computing Machinery, pp. 193–199. http://dx.doi.org/10.1145/1626195.1626244.

Swarnalatha, K.S., Ramchandra, K.C., Ansari, K., Ojha, L., Sharma, S.S., 2021. Real-time threat intelligence-block phising attacks. In: 2021 IEEE International Conference on Computation System and Information Technology for Sustainable Solutions (CSITSS). pp. 1–6. http://dx.doi.org/10.1109/CSITSS54238.2021.9683237.

T. N, N., Bakari, D., Shukla, C., 2021. Business e-mail compromise — techniques and countermeasures. In: 2021 International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE). pp. 217–222. http://dx.doi.org/10.1109/ICACITE51222.2021.9404587.

The Radicati Group, Inc., 2021. Email statistics report, 2021–2025. In: Tech. Rep. https://www.radicati.com/wp/wp-content/uploads/2020/12/Email-Statistics-Report-2021-2025-Executive-Summary.pdf.

Thomas, J., 2018. Individual cyber security: Empowering employees to resist spear phishing to prevent identity theft and ransomware attacks. Thomas, JE (2018). Individual cyber security: Empowering employees to resist spear phishing to prevent identity theft and ransomware attacks. International Journal of Business Management 12 (3), 1–23.

Volkamer, M., Renaud, K., Reinheimer, B., Kunz, A., 2017. User experiences of TORPEDO: TOoltip-poweRed phishing email DetectiOn. Comput. Secur. 71, 100–113. http://dx.doi.org/10.1016/j.cose.2017.02.004, https://linkinghub.elsevier.com/retrieve/pii/S0167404817300275.

Vrbančič, G., Fister, I., Podgorelec, V., 2020. Datasets for phishing websites detection. Data in Brief 33, 106438. http://dx.doi.org/10.1016/j.dib.2020.106438, https://www.sciencedirect.com/science/article/pii/S2352340920313202.

Wang, Y.-M., Beck, D., Wang, J., Verbowski, C., Daniels, B., 2006. Strider typo-patrol: Discovery and analysis of systematic typo-squatting. In: 2nd Workshop on Steps To Reducing Unwanted Traffic on the Internet (SRUTI 06). USENIX Association, https://www.usenix.org/conference/sruti-06/strider-typo-patrol-discovery-and-analysis-systematic-typo-squatting.

Weaver, B., Braly, A., Lane, D., 2021. Training users to identify phishing emails. J. Educ. Comput. Res. 59 (6), 1169–1183. http://dx.doi.org/10.1177/0735633121992516.

Zhu, E., Ju, Y., Chen, Z., Liu, F., Fang, X., 2020. [DTOF]-ANN: An artificial neural network phishing detection model based on decision tree and optimal features. Appl. Soft Comput. 95, 106505. http://dx.doi.org/10.1016/j.asoc.2020.106505, https://www.sciencedirect.com/science/article/pii/S1568494620304440.

Zhuo, S., Biddle, R., Koh, Y.S., Lottridge, D., Russello, G., 2023. Sok: Human-centered phishing susceptibility. ACM Trans. Privacy Secur 26 (3), 1–27.

Maxime Veit joined the SECUSO research group of Prof. Dr. Melanie Volkamer at the Karlsruhe Institute of Technology in November 2021. He received his Master’s degree in Information Systems from the Karlsruhe Institute of Technology. His research focuses on human interaction and risk communication in IT security in the context of web and email user interfaces.

Oliver Wiese is a postdoc at the CISPA Helmholtz Center for Information Security and works as a cybersecurity expert at European Digital Innovation Hub for AI and Cybersecurity (DAISEC) where he supports companies as well as the public sector in Germany in Cybersecurity. He completed his doctorate on phishing at Freie Universität in 2023 and has been researching human-centric cybersecurity since 2015.

Fabian Ballreich is a Research assistant in the SECUSO research group of Prof. Dr. Melanie Volkamer at the Karlsruhe Institute of Technology since July 2022. He graduated from the Karlsruhe Institute of Technology with a master’s degree in Information Systems. The overall goal of his work is the area of security awareness in the context of universities and organizations, i.e. in the field of information security incident reporting.

Prof. Dr. Melanie Volkamer Full professor of Security Engineering at the Karlsruhe Institute of Technology and head of the SECUSO research group since 2011. From August 2016 until March 2018 she was a Professor (Kooperationsprofessur) at the Department of Computer Science of Technische Universität Darmstadt. From December 2015 until December 2018, she has been appointed Full Professor for Usable Privacy and Security at Karlstad University. Before, she was an Assistant Professor at TU Darmstadt. From May 1st to Au-gust 31st 2011, she worked as a visiting researcher at CMU/CUPS. She has made contributions in the fields of web security and privacy, electronic voting, authentication and mobile security and privacy.

Douglas Engels is a Master student at the Freie Universität Berlin. He was member of the SCADACS team at the Freie Universität Berlin. The SCADACS was an organizational unit of the Secure Identity Research Group (AGSI) at Freie Universität Berlin, led by Prof. Volker Roth. It comprises research assistants and students of AGSI and student volunteers. He has a B.Sc. in Computer Science from the Freie Universität Berlin, and is at the time of writing finishing his M.Sc. in Computer Science at the same institute with a focus on computer security.

Peter Mayer is an assistant professor for usable security at University of Southern Denmark and KASTEL SRL fellow in the SECUSO research group at Karlsruhe Institute of Technology. He currently holds the roles of coordinator and co-speaker of the ‘‘Human and Societal Factors’’ research group and is a senior fellow of the Applied Computer Security Associates. He researches ‘‘End-user Viable Information Security & Privacy Solutions’’. Thereby, independently of whether end-users of security solutions are laypersons, admins, or developers, the focus lies on making security & privacy solutions viable for the target audience by considering their specific needs and skill sets.