PellucidAttachment: Protecting Users From Attacks via E-Mail Attachments

Sevtap Duman, Matthias Büchler, Manuel Egele, and Engin Kirda

***Abstract*—Malicious email attachments are a common and successful attack vector on today's Internet. Sophisticated attackers can craft highly-targeted attachments, using publicly available information about potential victims to create convincing documents that contain hidden malicious payloads. Users who open these attachments using vulnerable applications are at a high risk of infection. Unfortunately, current mitigations are unreliable, relying either on fallible malware detection techniques or user education. In this work, we propose adopting a default policy of isolated attachment rendering. Emails bearing attachments are transparently rewritten (in a sandboxed virtual machine environment) to contain static renderings of the attachments. Users who wish to obtain the original attachment are explicitly warned of the dangers of doing so – akin to TLS warnings as used in web browsers – before being allowed to access the requested documents. We implement this technique in a system we call PellucidAttachment . We further report on an extensive user study that measures the usability and effectiveness of PellucidAttachment in shielding users from attacks. Our evaluation shows that adopting email attachment security indicators and an isolation-by-default policy results in a significant increase in user security, while maintaining the usability of email attachments.**

***Index Terms*—Communications technology, communication systems, computer security, electronic mail, .Internet, Internet security, malware, message systems, phishing, software.**

I. Introduction

EMAIL is an essential communication tool. Email is used heavily for a wide range of activities such as sending out meeting invites, bills, receipts, and news articles. Often, documents are attached to emails, and the user is required to open this attachment to access the contents of the artifact.

Unfortunately, documents and links embedded in emails are a serious attack surface against users. Today, attackers exploit vulnerabilities in software that processes the content from these attachments to infect the targeted machine with malicious code. That is, once the victim opens the delivered attachment, an existing vulnerability (e.g., a use-after-free) can be exploited to execute arbitrary code on the victim's machine [1], [2].

Email-based attacks are often highly effective and successful. As a result, email is one of the main vectors for launching targeted attacks against specific victims. For example, it is widely reported that the Democratic National Committee was hacked using such targeted, spear phishing emails [3].

As email-based attacks are very successful in allowing attackers to compromise endpoints and gain an initial foothold for launching further attacks, this raises the question: What makes these attacks so successful in practice? The straightforward answer to this question is that users are typically not qualified to make security decisions regarding attachments they receive, often do not have updated systems, and often end up opening attachments that are highly risky. While deception techniques used by attackers such as persuasion, gain/loss framing [4] affect the success of a phishing email, emotional intelligence or salience, cognitive motivation, personality, and mood also play big roles in users' decision making process [5]. In fact, it is often difficult for a typical user to assess which attachments are riskier than others. That is, until an attachment has been downloaded and opened, a victim might not be able to easily determine if the attachment is a spear phishing attempt, or a legitimate artifact sent by someone that the victim knows. Hence, to check the contents of an attachment, a user is typically left with the sole choice of opening the attachment and attempting to read its contents.

Recognizing a malicious email might be difficult even for an expert user. While it is true that some emails might have traces of malicious behavior such as a suspicious-looking email sender or poor word choice in the subject [6], many malicious emails can appear very authentic. Although training users to spot phishing emails is helpful [7], spear phishing emails are very challenging to detect for most users. Particularly in attacks where the email sender imitates a trusted user, victims are prone to downloading and opening any attachments.

In spear phishing attacks, the attacker leverages information about the victim to tailor the attack email to improve the chance that the victim will click on the email attachment and open it. It has been reported that sophisticated targeted attacks (i.e., Advanced Persistent Threats (APTs)) often contain a spear phishing component [8]. Hence, it is clear that mechanisms are needed that can protect users against malicious attachments.

Existing solutions that use signatures and anti-virus scanning results rely on detection of malicious content before the delivery of the email attachments and leave the user vulnerable to

Manuscript received 24 March 2022; revised 3 March 2023; accepted 21 March 2023. Date of publication 22 May 2023; date of current version 16 May 2024. *(Corresponding author: Sevtap Duman.)*

This work involved human subjects or animals in its research. Approval of all ethical and experimental procedures and protocols was granted by the Northeastern University Institutional Review Board under Application No. IRB #16-09-08.

Sevtap Duman is with the Ege University, 35040 Izmir, Turkey, and also with the Northeastern University, Boston, MA 02115 USA (e-mail: sevtapduman@gmail.com).

Matthias Büchler and Manuel Egele are with the Boston University, Boston, MA 02115 USA (e-mail: matt.buechler.phd@gmail.com; megele@bu.edu).

Engin Kirda is with the Northeastern University, Boston, MA 02115 USA (e-mail: ek@ccs.neu.edu).

Digital Object Identifier 10.1109/TDSC.2023.3279032

undetected malicious content [9], [10], [11], [12]. Motivating from this we wanted to solve this issue by designing a user oriented approach where the user can view the contents of the attachments and make an informed decision before downloading the attachment.

In this paper, we propose a novel technical approach to protect users against malicious attachments. The important component of our approach is that converting the email attachment to an image format and attaching this image to the email, gives the user the opportunity to check the contents of an attachment without exposing themselves to malicious code. That is, users are able to peer into the contents of an attached document (e.g., a malicious PDF file) without having to download it, open it, and potentially be compromised. By converting potentially malicious files to a different format (e.g., converting a Word document to a PNG image), we remove the exploit code from the artifact and render it safe. The user can examine the contents and then interact with the original attachment only after having had a chance to check the authenticity and validity of the contents. This visual inspection prior to reaching the original content allows user to avoid downloading malware.

In order to evaluate the usability and effectiveness of our technique we sought to answer two research questions; RQ1: ”Does PellucidAttachment ’s rewriting capability prevent otherwise successful attacks?” and RQ2: ”Does PellucidAttachment improve the security of email users?” and we conducted a user study with 60 participants. The aim of the user study is to test the hypothesis that our proposed approach improves users’ security decisions. Our findings show that our proposed technique is a minimal addition to existing email security systems, and has significant security benefits for users in avoiding malicious attachments.

In summary, this paper makes the following contributions:

- We present a novel approach for protecting users against malicious email attachments that we call PellucidAttachment . Our proposed technique automatically renders attachments into safe PNG images, and replaces the original attachment with the generated image. The conversion gives users the chance to distinguish between a benign attachment and a malicious one without having to open the attachment and potentially be compromised.
- We empirically evaluated our approach with 39 real-world malicious attachments. We show that by rendering malicious attachments into PNG images, our system removes the existing exploit code for all of the tested files (10 PDF, 10 Microsoft Excel, 10 Microsoft Word, and 9 PNG files).
- We evaluated the security benefits of our approach with a randomized user study (n = 60). Our multi-protocol user study shows that PellucidAttachment is usable and improves user security by helping them avoid exposure to malicious documents.

To further the spirit of open science, we will release our implementation of PellucidAttachment under an open source license.

The remainder of this paper is structured as follows. In Section III, we provide background information on malicious email attachments. Section II discusses related work. In Section IV, we present the overview of our proposed approach. In Section V, we discuss our threat model and our assumptions. Section VI describes a prototype implementation of our approach. Section VII presents an evaluation of the proposed system with real users and, finally, Section IX concludes the paper.

II. Related Work

Research studies related to email attachment security mostly focus on detection of malicious content at the spam or antivirus scanning layer or they only look at increasing user awareness by phishing training. In this section we covered the published work related to protection methods from malicious email attachment.

Malicious PDF files can be created by embedding JavaScript, executable code, or any other content directly into the PDF. One of the most commonly used techniques to detect such attacks is structural analysis (e.g., checking n-gram features, the number of objects and the streams of the PDF file, etc.). Laskov and Šrndić [13], Smutz and Stavrou [14], and Šrndić and Laskov [15] perform structural analysis of PDF files to assess if the file is malicious. Other research groups, in contrast, have used reverse mimicry techniques to show that assessing structural features alone is not enough [16] for detecting malicious documents.

Liu et al. [17] present a different approach to detect malicious PDF files. The authors use both static and dynamic features for detection, and implement a prototype malicious PDF detector. They evaluated their system with real benign and malicious samples. These solutions rely on JavaScript exploitation of PDF files and ignore other exploitation techniques and other filetypes leaving users vulnerable to wide range of malicious files.

In 2001, Balzer [10] implemented a system called SafeEmailAttachments as a wrapper on Windows NT systems. SafeEmailAttachments was designed to follow safety and active content rules before the attachment was authorized to be opened. SafeEmailAttachments successfully blocked the I-Love-You virus when the virus first started spreading [18]. However, it is limited by specific operating system and email client.

One of the earliest malicious file detection studies based on n-grams is MEF; Malicious Email Filter [12]. In this work, authors introduce byte-sequences as a feature set to train their model. Since then n-grams analysis has been widely used in malicious file detection including MEADE [9] which is a recent study. In MEADE, authors collect malicious Microsoft Office document and Zip archive data from VirusTotal. They use deep neural networks (DNN) and gradient boosted decision tree ensembles to detect malicious email attachment. DNN model is able to detect 5 out of 9 Petya samples.

Another early studies of decision theory approach on email security is conducted by Dong-Her et al. [11]. In their work, they utilize a popular probability model Bayesian Network to detect malicious emails. They include a discussion section specifically on management of email where they mention common human behavior and social engineering. Our approach does not use any classification methods to find malicious or benign files, as a result does not have any false positive or false negative results.

Human effect in security vulnerabilities has been investigated from a social engineering point of view and user training models have been suggested. Dodge et al. [19] constructed phishing emails and used these phishing emails as part of their user training to increase user awareness. With their unannounced phishing email attacks over two years, they have seen increased security awareness and decrease in providing sensitive information. In their study, Goel et al. [4] look into how contextualized emails affect susceptibility of users in phishing email tests. Oliveira et al. [5] examined deceptive cues that make messages more appealing to users. As a result of their study they claimed that user awareness is crucial to mitigate phishing effectiveness. To raise awareness, U.S. The Federal Bureau of Investigation published articles on Spoofing and Phishing [20] and Business Email Compromise [21] where attackers send emails to victims pretending to be from someone they would know such as a colleague or boss. During these attacks the victim is persuaded the email originated from a legitimate source and they act upon the email to provide requested action in the email.

Malicious Email Tracking (MET) addresses the virus infection problem through email by using behavioral-based analysis [22]. Later, the authors proposed an approach that supports a wider scope of this online behaviour-based security system [23].

Muniandy et al. [24] proposes a practical approach to educating Internet users using email screenshots. To increase the awareness of phishing emails, screenshots of dedicated phishing emails are shown to the Internet user. These screenshots highlight characteristics upon which a user can recognize phishing attempts. Both our and Muniandy's approaches leverage visual impressions to let the user decide if the email is benign or malicious. However, our approach differs in several fundamental ways. First, our approach is not primarily for educational purposes. Second, our tool processes every incoming mail. Third, our system generates an image for every single email attachment whereas Muniandy uses eight pre-defined dedicated screen shots for educating people.

Studying the effectiveness of security warning designs in the context of Human Computer Interaction (HCI) has been a particular focus of usable privacy and security. The authors, Petelka et al. [25], conducted a user study to compare the effectiveness of different warning designs in preventing users from clicking on phishing links in emails. The most effective method found in the study was forcing attention to the warning by deactivating the original link. In their study, Jaeger et al. [26] used eye tracking and a post-experimental survey to assess how users collect security-related information cues. They have observed that situational information security awareness is positively impacted by a security warning. In 2021, Gutfleisch et al. [27], conducted a series of experiments to evaluate the effectiveness of different MS Office macro warning designs in preventing users from running malicious macros. One of the outcomes of the study was that the design of warning messages could mislead users and this may be a significant factor in why macros are frequently enabled. They have suggested conducting more usability tests of security features.

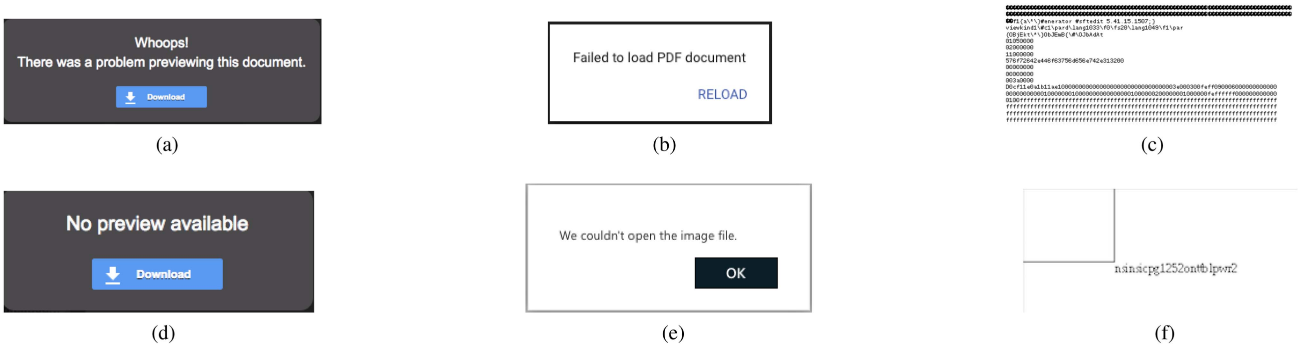
Moreover, there are commercial solutions for preventing virus distribution through email that integrate an anti-virus engine into their MTA [28], [29]. These commercial tools claim to provide an image display of the delivered attachments similar to GMail [30] and Outlook [31]. However, we could not gather any information about these commercial solutions to perform a comparative analysis. In order to compare rendering capacity of PellucidAttachment to these commercial tools we used the same malicious dataset obtained from VirusTotal. As shown in Fig. 1, PellucidAttachment successfully revealed the content of the malicious attachments. Fig. 1 demonstrates that both GMail and Outlook fail to display any preview of the tested malicious files and provide any guidance to users.

Additionally, there is a variety of research on isolation of execution environments for preventing attacks through email or web browsers where users can download a malicious file in a virtual machine. For example, Moshchuk et al. [32] present an anti-malware tool called SpyProxy. This tool detects drive-by-download attacks by rendering webpages in virtual machines. SpyProxy was developed as a front end module that redirects HTTP requests to a virtual machine depending on the webpage contents. The webpage undergoes static analysis, and if the webpage has active content or has any non-HTML content types that are considered to be unsafe, the page is queued for dynamic processing. A performance analysis of SpyProxy revealed that it would add a considerable (600 ms) delay to each web page load. Furthermore, Radhakrishnan et al. [33] propose to leverage dynamic sandboxing to provide an isolated execution environment for potentially malicious content. These works drastically differ from PellucidAttachment, along two major directions. First, users can benefit from PellucidAttachment without changing the workflow or tools (e.g., mail clients) they are already accustomed to, and these tools do not need to be modified. Secondly, while virtual machines and sandboxes are at the core of the above-mentioned systems, PellucidAttachment merely uses virtual machines exclusively for the purpose of automatically rendering attachments into image files. A PellucidAttachment user is not inconvenienced by the existence of a virtual machine, nor is she even aware of its use.

In summary, our proposed solution is substantially different from existing research. Our solution PellucidAttachment is user oriented that a user can view the contents of an attachment safely and make informed decisions but it is not designed solely for educational purposes. Unlike previous work, PellucidAttachment does not try to distinguish between malicious or benign attachments. Instead, our approach converts attachments into images so that the user has an opportunity for safe decision making process without exposure to malicious content.

III. Background

Many document formats, such as MS Word, MS Excel, PDF, and PNG documents, can be crafted to be malicious. Once exploit code is inserted into the document, malicious activity can be triggered, often just by opening and viewing the document. Document viewers or editors may be vulnerable to memory corruption exploits due to unpatched or zero-day security flaws,


Whoops!
There was a problem previewing this document.

Failed to load PDF document

```
f1{a\*\}#enerator #sftedit 5.41.15.1507;}
viewkind1\#c1\pard\lang1033\f0\fs20\lang1049\f1\par
{OBjEkt\*\}ObJEmB{\#\OJbAdAt
01050000
02000000
11000000
576f72642e446f63756d656e742e313200
00000000
00000000
003a0000
D0cf11e0a1b11ae10000000000000000000000000000000003e000300feff090006000000000000000
0000000000001000000010000000000000000010000002000000010000000feffffff0000000000000
0100ffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff
ffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff
ffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff
ffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff
```

No preview available

We couldn't open the image file.

nsinsicpg1252ontfblpwr2

Fig. 1. A malicious PDF file and a malicious PNG file were examined through different MUAs. Figure 1a is the preview of a malicious PDF file through GMail and 1b is the preview of the same malicious PDF file through Outlook. Figure 1c is the first a few lines of a PNG preview of a malicious PDF file. Similarly, a malicious PNG file delivered using GMail produced Figure 1d, and using Outlook produced Figure 1e. When the malicious PNG file was sent as an email attachment processed by PellucidAttachment, the user was delivered a rendered version of the original attachment shown in Figure 1f.

while some document formats such as PDF can also potentially contain malicious scripts.

A. Document Vulnerabilities and Exploitation

1) Attacks Via Microsoft Office Files: An attacker may be able to craft a malicious Microsoft Office file that runs arbitrary code when the document is opened. Some unpatched versions of Microsoft Office have memory corruption, elevation of privilege, denial of service, and similar vulnerabilities. A recent example is CVE-2016-7193 [34], where RTF file content is not handled properly by the software, leading to the execution of attacker-supplied code.

Macros are another popular method that attackers use to launch attacks. Macros are used to simplify common tasks by automating them in Microsoft Office. However, this legitimate functionality may be used to deliver malware as well. In their paper Dechaux et al. [35] presented how attackers can create new documents with malicious macros and bypass existing detection mechanisms. Unfortunately, macros have been abused by attackers so often [36] that they have been disabled in recent versions of Microsoft Office. Nevertheless, an attack may be successful if the victim chooses to run the macro (e.g., through social engineering).

2) Attacks Via PDF: A vulnerability in a PDF reader may cause arbitrary code to be executed on the targeted host. The complex structure of PDF files has historically provided attackers many opportunities to exploit memory corruption errors. PDF documents may also be able to run unauthorized JavaScript, ActionScript, and other types of malicious scripting code.

An example of a recent Adobe PDF vulnerability [37] allows remote attackers to execute arbitrary code on vulnerable installations of Adobe Acrobat Reader DC [38]. As examples of PDF attacks, Mimicus and reverse mimicry attacks are trying to hide the malicious content from a PDF malware detector by using machine learning techniques [39]. Where mimicus is trying to change the content of a malicious pdf file to match a benign PDF file's features, in reverse mimicry the attacker hides malicious content in a benign file trying to make minimum changes.

3) Attacks Via PNG: Imagemagick [40] is a free software package that allows developers to programmatically manipulate images. As a result of its advanced capabilities, attackers may be able to craft PNG images that are malicious. For example, a recently exposed vulnerability was published on the Common Vulnerabilities and Exposures Database where an attacker can execute arbitrary code via shell meta-characters in a crafted image [41].

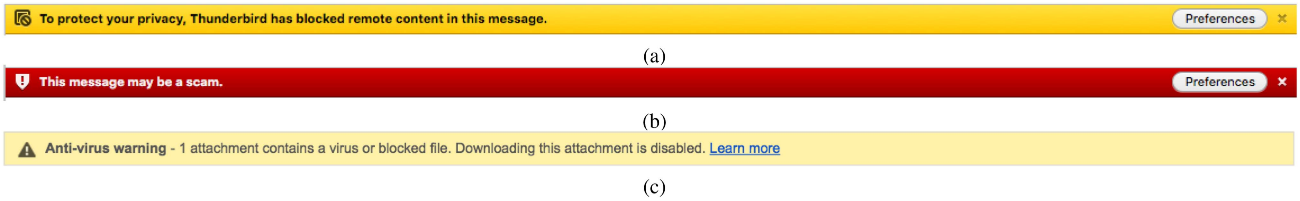
Moreover, the PNG reference library libpng [42] is also vulnerable to various memory corruption attacks. CVE-2016-10087 [43] is an example of the libpng vulnerability, where the attacker takes advantage of a null dereference bug in earlier versions of libpng. In 2016 Stegano/Astrum, DNSChanger and Sundown exploits used PNG files to cloak their exploits [44]. While Stegano/Astrum and DNSChanger are used mainly in malvertising, Sundown is used to hide either the stolen info or the exploit code.

B. Defense Against Malicious Files

The best defense mechanism against malicious email attachments would be to prevent them from being downloaded to the victim's system. However, this would require that benign documents can reliably be distinguished from malicious ones. Static or dynamic analysis techniques may be used to perform this detection.

Unfortunately, as explained in Section II, static and dynamic detection techniques have their limitations. As a result, once a malicious document falls through the cracks, the user needs to make a decision. In fact, in most of the attacks listed in the previous section, the attacker counts on the victim's input such as downloading the malicious email attachment, opening a PDF file that launches a remote attack via JavaScript code, activating the macros of a Microsoft Office document, and viewing a PNG image that opens a backdoor on the compromised system.

Previous research has determined that warning users frequently about the results of their actions may be effective [45] in detecting some attacks. In the case of email security warnings, mail user agents (MUA) (i.e., email clients) have improved over the years. Two concrete examples of MUA warnings would be the warning banners that Thunderbird and GMail present to users. Such banners inform users whether the content of the email is suspicious (Fig. 2). However, these banners are



To protect your privacy, Thunderbird has blocked remote content in this message.

(a)

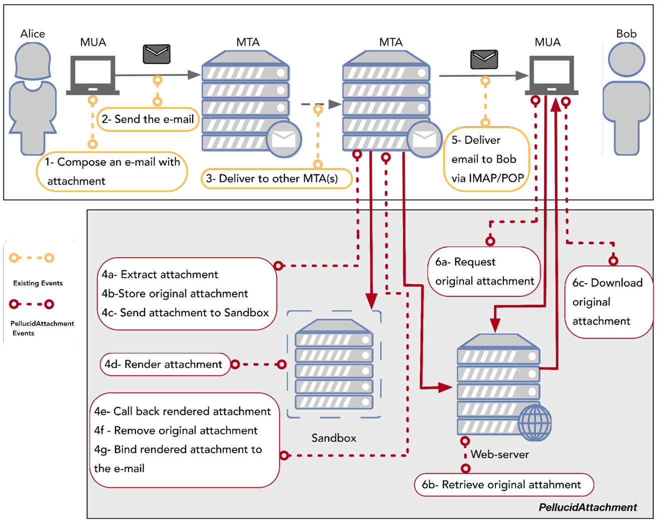
This message may be a scam.

(b)

Anti-virus warning - 1 attachment contains a virus or blocked file. Downloading this attachment is disabled. Learn more

(c)

Fig. 2. Email warning banners. In Figure 2a, a security alert banner is shown to Thunderbird users when there is an image or stylesheet embedded in an email message. Similarly, in Figure 2b another banner is shown to notify users about a suspicious email which might be a potential phishing attempt. Figure 2c shows a security banner that blocks users from downloading a possibly malicious file.



Alice
MUA
MTA
MTA
MUA
Bob
2- Send the e-mail
1- Compose an e-mail with attachment
3- Deliver to other MTA(s)
5- Deliver email to Bob via IMAP/POP
Existing Events
PellucidAttachment Events
4a- Extract attachment
4b-Store original attachment
4c- Send attachment to Sandbox
4d- Render attachment
4e- Call back rendered attachment
4f - Remove original attachment
4g- Bind rendered attachment to the e-mail
Sandbox
6a- Request original attachment
6c- Download original attachment
Web-server
6b- Retrieve original attahment
PellucidAttachment

Fig. 3. Overview of the system.

unfortunately frequently ignored by users due to the underlying detection algorithms' lack of precision (i.e., too large a false positive rate) [46], [47].

In this paper, rather than trying to detect malicious files that are spread through email, we adapt a generic defense approach that converts the potentially malicious document to a harmless image format. By doing so, we automatically remove and prevent the exploit.

IV. System Overview

The key insight of our work is that email users have insufficient information to distinguish potentially malicious attachments from benign attachments. PellucidAttachment narrows this information gap by allowing the user to safely peek into the contents of each attachment without first downloading and opening it. Thus, if the provided information allows users to make better judgments, the security posture of email users is improved. PellucidAttachment provides this capability by modifying emails as they are received at the recipient's mail transfer agent (MTA). In particular, PellucidAttachment modifies how the MTA processes incoming emails along two dimensions (Fig. 3). First, attachments of incoming emails are *replaced* by renderings thereof. Second, the system provides a mechanism to access the *original* unmodified attachments if the user so chooses.

A. Replacing Attachments

To replace incoming email attachments, PellucidAttachment follows a sequence of three consecutive steps for each attachment.

1) Extract and Preserve Original Attachment: Upon receipt of a new email, PellucidAttachment parses the email content and extracts all attachments. Each attachment is then used in two ways. First, PellucidAttachment persists the attachment in case the user needs access to the unmodified attachment later on (see Section IV-B). Second, each attachment is subjected to a conversion process where its contents are rendered into an image.

2) Render Attachment Into an Image: PellucidAttachment converts each attachment into a visual representation (i.e., an image) of its content. For example, the contents of a PDF document will be rendered into an image file that visually carries the same information as the *original* file itself. Note that the input to this rendering process are the potentially malicious attachments sent by the attacker. Thus, this conversion step warrants additional security precautions. To counter the situation where a malicious attachment attacks and exploits PellucidAttachment's rendering infrastructure, all conversion is performed in a sandboxed virtual machine that is restored to a known good state for each attachment. Furthermore, PellucidAttachment provides a firewall around the sandbox to prevent any communication beyond the MTA and the sandbox itself. Thus, for each attachment, PellucidAttachment requests the sandbox to convert the attachment into an image, and then retrieves the image from the sandbox.

3) Replace Original Attachment With Its Image: The final step carried out by the MTA replaces the original attachments with the rendered images thereof. In addition to replacing the attachments, PellucidAttachment also includes a link at the bottom of each email that allows the user to access the original unmodified attachment if needed.

B. Accessing Original Attachments

Sometimes, the user must access the original, unmodified attachment that was included in the email. This can become necessary if the user, for example, is required to fill a form or is expected to make modifications to a document. A user can gain access to the unmodified attachments by following the link at the bottom of the email. However, instead of providing direct access to the unmodified and potentially dangerous attachments, PellucidAttachment confronts the user with a security warning analogous to TLS certificate warnings used by all major web browsers. Thus, before gaining access to the original attachments, users have to acknowledge their awareness of potential negative impacts.

V. Threat Model

As PellucidAttachment tries to protect recipients from email messages that include malicious attachments, we assume the following threat model. First, we assume that the attacker has a reliable way of delivering malicious emails to the victim. That is, the attacker has the capability to circumvent all of today's frequently deployed defensive measures, such as spam filtering, anti-virus scanning of email attachments, or statistical models to detect malicious emails. Furthermore, the attacker knows about the software installed on the victim's computer, and additionally knows of at least one arbitrary code execution vulnerability in one of the installed software packages. In addition to the vulnerability, the attacker has the capability to create exploits that target that vulnerability and include this exploit in a file of the format that will be processed by the vulnerable software if the victim opens the file.

A concrete and realistic example is an attacker with knowledge of a vulnerable version of Adobe Reader installed on the victim's machine, and a readily available exploit in the form of a malicious PDF file that will grant the attacker arbitrary code execution capabilities if it were to be opened with the vulnerable software.

Beyond these capabilities, the attacker is also assumed to be aware of PellucidAttachment and its use by the victim, and he might want to attack PellucidAttachment itself instead of the victim user. Note that PellucidAttachment 's main goal is to provide additional information about attachment content without exposing users to the potential threats therein. However, PellucidAttachment must and does provide the user with access to the original attachments if the user so chooses. Thus, while we assume that the attacker has the various technical capabilities outlined above, we also assume that the attacker does not have the capability to lure the user into downloading and opening the *original* malicious attachment. This assumption is realistic when considering an attacker who indiscriminately attacks his victims. We argue that creating emails that convince users to download an attachment *despite the provided preview*, acknowledge the security warnings, and then open the resulting file, requires significant and more importantly individualized effort, and thus significantly raises the bar for the attacker.

While operating in the above-stated threat model, PellucidAttachment tries to impose as few restrictions on users as possible and thus is deployed exclusively at the MTA of the recipient. This implies that users can continue using the mail user agents (MUAs) that they are most accustomed to without any changes to the client-side software. Furthermore, as PellucidAttachment operates in conjunction with the MTA, it is compatible with an enterprise setting where a company maintains its own email system. Deployed in this way, PellucidAttachment can seamlessly afford its protections to any user throughout the enterprise.

VI. Implementation

We implemented a prototype of PellucidAttachment on Ubuntu Linux running Postfix [48] as the MTA. PellucidAttachment introduces three additional components to an existing MTA: a content filter, the rendering sandbox, and a facility to provide access to unmodified attachments. We provide detail on each of these components in the following.

A. Content Filter

Postfix uses the term *content filter* for any software component that inspects or modifies email data (including both headers and payload). To simplify the process of creating content filters, Postfix provides a standardized interface that PellucidAttachment leverages. The MTA is configured to trigger the PellucidAttachment filter which in turn parses the body of each email.

The content filter first extracts all attachments and preserves them should the user require access to them later on (see Section II). Subsequently, the attachments are sent to the rendering sandbox that converts each attachment into a visual representation thereof.

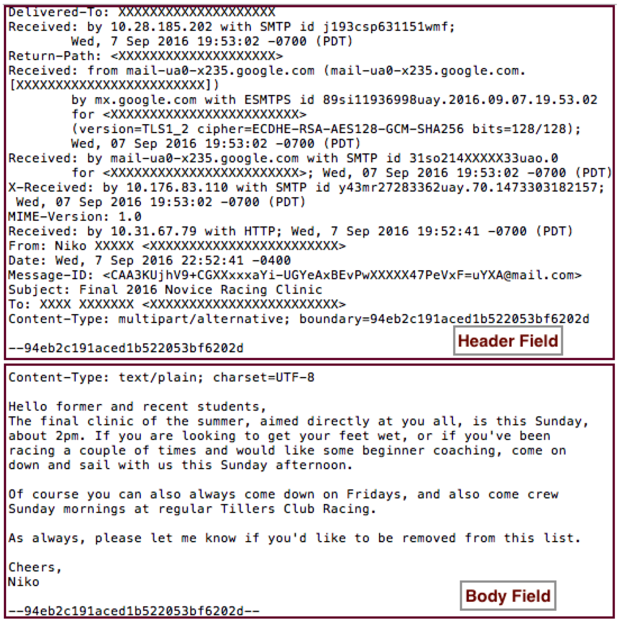
Finally, the content filter replaces the original attachments with the renderings obtained from the sandbox, and inserts links at the bottom of the email to allow the user to fetch the unmodified attachments.

To extract attachments from an email, the content filter parses the information contained in the message. Within a MIME email message, individual attachments are described through a content-disposition header field according to the specification in RFC2183.[1] For example, to transmit UTF-8 formatted data through the ASCII SMTP protocol, the content needs to be identified and encoded appropriately. Fig. 4 shows an example where the UTF-8 formatted text of an email is labeled with a corresponding content type. Thus, PellucidAttachment iterates over all attachments identified in this way and stores the original content locally to allow the user to retrieve the original attachment should that be necessary. Note that for security purposes, PellucidAttachment assigns new random file names when storing the attachments locally. This is similar in spirit and motivation to the sharing capabilities of systems such as Google Drive or Dropbox. The long random file names represent a capability that prevents attackers from enumerating or iterating over all attachments stored on the server. Moreover, these operations communicated through a trusted and privacy-preserving infrastructure. As the next step, the content filter forwards each attachment to a sandbox to render its contents into an image.

B. Rendering Sandbox

The goal of the rendering sandbox is to convert a given attachment into a visual representation (i.e., an image) of its content.

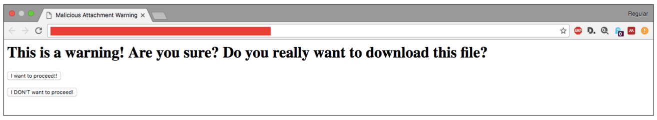
[1] [Online]. Available: https://tools.ietf.org/html/rfc2183



```
Delivered-To: XXXXXXXXXXXXXXXXXXXX
Received: by 10.28.185.202 with SMTP id j193csp631151wmf;
        Wed, 7 Sep 2016 19:53:02 -0700 (PDT)
Return-Path: <XXXXXXXXXXXXXXXXXXXX>
Received: from mail-ua0-x235.google.com (mail-ua0-x235.google.com.
[XXXXXXXXXXXXXXXXXXXXXXXX])
        by mx.google.com with ESMTPS id 89si11936998uay.2016.09.07.19.53.02
        for <XXXXXXXXXXXXXXXXXXXXXXXX>
        (version=TLS1_2 cipher=ECDHE-RSA-AES128-GCM-SHA256 bits=128/128);
        Wed, 07 Sep 2016 19:53:02 -0700 (PDT)
Received: by mail-ua0-x235.google.com with SMTP id 31so214XXXXX33uao.0
        for <XXXXXXXXXXXXXXXXXXXXXXXX>; Wed, 07 Sep 2016 19:53:02 -0700 (PDT)
X-Received: by 10.176.83.110 with SMTP id y43mr27283362uay.70.1473303182157;
 Wed, 07 Sep 2016 19:53:02 -0700 (PDT)
MIME-Version: 1.0
Received: by 10.31.67.79 with HTTP; Wed, 7 Sep 2016 19:52:41 -0700 (PDT)
From: Niko XXXXX <XXXXXXXXXXXXXXXXXXXXXXXX>
Date: Wed, 7 Sep 2016 22:52:41 -0400
Message-ID: <CAA3KUjhV9+CGXXxxxaYi-UGYeAxBEvPwXXXXX47PeVxF=uYXA@mail.com>
Subject: Final 2016 Novice Racing Clinic
To: XXXX XXXXXXX <XXXXXXXXXXXXXXXXXXXXXXXX>
Content-Type: multipart/alternative; boundary=94eb2c191aced1b522053bf6202d

--94eb2c191aced1b522053bf6202d
```

```
Content-Type: text/plain; charset=UTF-8

Hello former and recent students,
The final clinic of the summer, aimed directly at you all, is this Sunday,
about 2pm. If you are looking to get your feet wet, or if you've been
racing a couple of times and would like some beginner coaching, come on
down and sail with us this Sunday afternoon.

Of course you can also always come down on Fridays, and also come crew
Sunday mornings at regular Tillers Club Racing.

As always, please let me know if you'd like to be removed from this list.

Cheers,
Niko

--94eb2c191aced1b522053bf6202d--
```

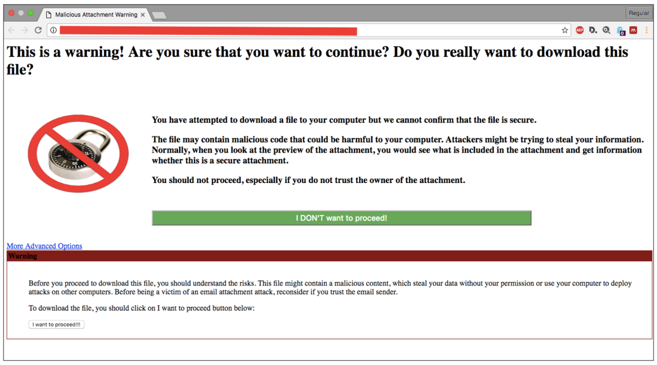
Fig. 4. Raw format view of an Email.



[redacted]

This is a warning! Are you sure? Do you really want to download this file?

Fig. 5. Warning page version 1.



[redacted]

This is a warning! Are you sure that you want to continue? Do you really want to download this file?

You have attempted to download a file to your computer but we cannot confirm that the file is secure.

The file may contain malicious code that could be harmful to your computer. Attackers might be trying to steal your information. Normally, when you look at the preview of the attachment, you would see what is included in the attachment and get information whether this is a secure attachment.

You should not proceed, especially if you do not trust the owner of the attachment.

Before you proceed to download this file, you should understand the risks. This file might contain a malicious content, which steal your data without your permission or use your computer to deploy attacks on other computers. Before being a victim of an email attachment attack, reconsider if you trust the email sender.

To download the file, you should click on I want to proceed button below:

Fig. 6. Warning page version 2.

Because the attacker might try to attack PellucidAttachment directly, the rendering sandbox operates in a virtual machine environment.

Our prototype implementation uses the KVM [49] hypervisor for this purpose. In a naive implementation, PellucidAttachment would spawn a new virtual machine from a known clean state for each attachment. However, to optimize performance without compromising on functionality, PellucidAttachment does not boot the virtual machine instance from a power-off state, but rather uses the snapshotting mechanism provided by the KVM hypervisor.

Of course, PellucidAttachment can only render attachments into images if the attachment is of a known file type. To achieve compatibility with a large number of file formats that are frequently used as email attachments, PellucidAttachment leverages off-the-shelf utilities such as the LibreOffice [50], Ghostscript [51], and Imagemagick [40]. Thus, PellucidAttachment supports any format produced by Microsoft Office products, PDF and postscript content, and dozens of image formats.

Our current prototype implementation renders each attachment into an image in the PNG file format. As all popular mail user agents support PNG files natively, this ensures compatibility with a wide user base. Once the attachment is rendered, the content filter receives the resulting image and continues to modify the content of the email.

C. Replacing Attachments

To replace the converted attachments, the content filter simply replaces the content of the original attachment with the resulting images obtained from the rendering sandbox. While removing the original attachments is straightforward, PellucidAttachment must take care to insert the new images with the correct meta-information to describe the content type, length, and encoding. The filename of the converted attachment is assigned after the *original* attachments' name (including the old extension), followed by the page number and new extension. Therefore, the recipient of the email can use the filename information of the file beside the image version of the attachment to decide the validity of the email attachment before downloading the *original* attachment.

In addition to inserting the rendered images, PellucidAttachment also modifies the content of the email to include hyperlinks to the unmodified attachments stored on the mail server. Of course, these links correctly refer to the randomized file names mentioned above.

D. Providing Access to Unmodified Attachments

Should users require access to unmodified attachments, PellucidAttachment serves the unmodified attachments via HTTP. To this end, our prototype leverages the popular NGINX [52] HTTP server. Of course, it is straightforward to use any other HTTP server such as Apache or lighttpd instead.

To request access to an unmodified attachment, the user simply follows the corresponding link that PellucidAttachment inserted at the bottom of the email. However, although attachments are served by a web server, the server prevents direct access to the content and instead presents the user with a warning screen (see Figs. 5 and 6) modeled after the TLS certificate warnings found in all major web browsers. This warning informs the user of the potential negative implications of accessing the original attachment, and serves as a deterrent to unnecessary exposure to malicious documents. Only after the user has acknowledged that they want to proceed to download the original attachment does the download begin.

VII. Evaluation

In this section, we present the experiments and results we obtained by evaluating PellucidAttachment along two orthogonal dimensions that aim to answer the following two research questions.

RQ1: Does PellucidAttachment 's rewriting capability prevent otherwise successful attacks? PellucidAttachment aims to improve the security of email users by rewriting email attachments with rendered images thereof. As the resulting images are included in the email, PellucidAttachment must ensure that any malicious components of the original attachments are filtered out during the rendering process. To demonstrate the technical efficacy of PellucidAttachment against successful attacks through email attachments, we evaluated PellucidAttachment with a variety of malicious files.

RQ2: Does PellucidAttachment improve the security of email users? The rewriting capabilities implemented by PellucidAttachment provide additional information to the recipient of an email. Ideally, this additional information allows the users of our system to make better security decisions (i.e., whether they should open a given email attachment or not). To answer this research question, we performed an extensive user study on 60 volunteers.

In summary, our evaluation found that PellucidAttachment answers RQ1 in the affirmative and demonstrates significant (i.e., almost 4x) improvements of the security of email users against malicious attachments as an answer to RQ2. The details of these experiments are presented next.

A. RQ1: Efficacy of PellucidAttachment

The technical efficacy of PellucidAttachment is determined by the system's capability to replace malicious attachments with benign renderings of their contents. Thus, to evaluate this aspect of PellucidAttachment, we obtained a representative sample of malicious files whose file types correspond to those commonly used in email-borne attacks. We obtained our dataset of malicious files from Google's VirusTotal service [53]. In total, our dataset consisted of 39 malicious files (10 `.pdf`, 10 `.docx` (i.e., MS Word), 10 `.xlsx` (i.e., MS Excel), and 9 `.png`).

Once the files were confirmed as malicious, we composed one email per file and included the file as an attachment in the email. These emails were sent to an MTA that ran the PellucidAttachment system. And we tested the effectiveness of PellucidAttachment by passing each one of the malicious file in our dataset through our system. To ascertain whether PellucidAttachment indeed strips the malicious functionality from replaced attachments, we opened each email with Thunderbird. As expected, we did not observe any signs of a malware infection after the attachments had been rewritten by PellucidAttachment. Additionally, we submitted all rewritten attachments to VirusTotal and not a single alert was raised, nor was any of the submitted files labeled as suspicious.

B. RQ2: Security Improvements for Email Users

To assess whether PellucidAttachment improves the security of regular email users, we performed the following user study. As elaborated above, the rewriting capabilities offered by PellucidAttachment add information to the email that a user can leverage when considering whether she should open a given attachment or not. We consider that PellucidAttachment increases a user's security if this additional information is sufficient for the user to decide not to open otherwise malicious email attachments.

Thus, to evaluate PellucidAttachment 's effectiveness in this regard, we designed a test scenario where participants were asked to read a series of emails. As the user study involves human volunteers as test subjects, we applied for and obtained approval from our university's institutional review board prior to launching the experiments.

1) Study Design: Participants were initially unaware of the purpose of the study. Instead, participants were told the study was an "experiment to investigate the effects of interruptions on concentration and decision-making when multi-tasking." Each participant was debriefed and informed of the true goal of the study once she finished the experiment.

2) Experiment Environment: Each participant in the study was provided with access to a Windows computer where Microsoft Office, Adobe Reader, and a pre-configured installation of Mozilla Thunderbird as the MUA were installed. The entire experiment was conducted in the context of the following fictitious scenario. We instructed study participants that they should assume the role of a graduate student aide in the university's writing center. As one would expect, this role included tasks such as answering emails and reviewing and editing documents (e.g., for grammar and spelling) that other students at the university submitted (also via email). Furthermore, participants were told that they should assume that the provided email account was their private university account, and thus treat it with equal care as their real accounts. Finally, the experiment suggested the prospect of a PayPal gift card for exceptional service.

Once the scenario was set up, each participant received a sequence of 16 emails. As this study focuses on email attachments specifically, the email set was structured as follows. Out of the 16 emails, 10 had no attachments, 4 emails featured malicious attachments, and the attachments of the remaining 2 emails were benign. With this distribution we aimed to simulate real life experience of email flow for the test individuals. The focus of our attention was on whether users would open the 4 malicious attachments, and whether the introduction of a system such as PellucidAttachment would have a positive effect on this number (i.e., fewer opened malicious attachments). To identify whether users opened attachments, all interactions with the provided computer were screen-captured and evaluated by the authors. True to the study protocol, all interaction between the researchers and the participants happened via email.

3) Recruitment: We recruited participants on our university campus and the broader metropolitan area via email that explains our study. In the recruitment email, we stated that participants do not need to share their private information to participate and we did not collect any personally identifiable information as part of the university's institutional review board risk and confidentiality procedures. We only applied two criteria to prospective study participants. First, all applicants had to be between 18 and 49 years of age, and second, computer science students were excluded from the study. We excluded members from the computer science department to prevent any sort of security-knowledge bias that might be ingrained in CS students. Following this process, we collected 60 volunteers from a broad spectrum of occupations, backgrounds, ethnicities, nationalities, and gender. The study participants consisted of 45% students, 28.3% research assistants, 8.3% postDocs, 3.3% doctors, 3.3% engineers and the remaining 12% with various occupations 9. We did not collect any Personally Identifiable Information (PII) that can be used to identify user directly or indirectly.

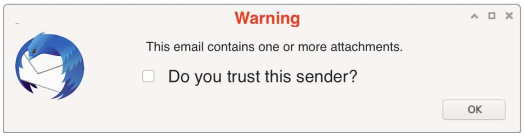
4) Experiment Details: On average, each participant required 25 minutes to read and react to all of the 16 emails. Additionally, each participant spent around 10 minutes to complete the study protocol form, the consent form, and answer the security awareness questionnaire detailed later in this section.

Of the 16 emails, the first and last email had the purpose of introducing participants to the system, and informing them that the experiment had concluded. The three emails following the introductory email consisted of two emails with benign attachments and one email without an attachment. The remaining eleven emails were a random sequence of the four emails with malicious attachments and the remaining seven emails with no attachments.

The four emails that contain malicious attachments were modeled after real-world attacks as follows. One of the attack emails imitated a non-existent university division that invited students to register for courses and internships by submitting the attached document. Another attack email was composed to imitate a PayPal notification and allegedly included an attached invoice. The third attack email appeared to originate from the university's human resources division and featured a subject line of "documents from work." The email did not contain any text but included a PDF attachment with a filename of "everyones_updated_salary_chart.pdf". The fourth attack email was designed to trick users into sharing their password and deleting their emails. The email was tailored to imitate an email from the IT department. It included a file named "account_confirmation", and the users were told that their mailbox exceeded the storage limit. The email explained that if the users would like to continue receiving email, they should fill out the attached document with their username, password, and an error code that is inserted at the end of the email, and send the document back to the IT department.

5) Protocols: We assigned the 60 participants in our study to four distinct groups. The control group and Groups 0, 1 and 2 contained ten participants each. The remaining 20 participants were assigned to Group 3.

The control group and Group 0 followed the study protocol without the protections afforded by PellucidAttachment, but Group 0 had a warning pop-up for the emails included attachments. The differences between Groups 1, 2, and 3 were confined to the introductory email and the design of the warning page. As our results demonstrate, slight variations in the introductory email or the warning page can have positive effects on the realized security gains. We included the control group in our experiment to establish a baseline against which we can measure the improvements introduced by PellucidAttachment.



Warning

This email contains one or more attachments.

□ Do you trust this sender?

Fig. 7. Warning pop-up.

TABLE I
PROTOCOL DESIGNS

Label	Introduction Email Version	Warning Page Version
Control	N/A	N/A
Group 0	N/A	N/A
Group 1	Version 1	Version 1
Group 2	Version 1	Version 2
Group 3	Version 2	Version 2

Participants in Group 0 asked to verify the email sender before downloading attachments whenever then received an email with attachments. The warning pop-up (see Fig. 7) blocks the user to download the attachment unless they state that they trust the sender.

Participants in Group 1 received their emails through PellucidAttachment. The introduction email for Group 1 contained a description of the differences in the style of email the participants would receive. This email described that any attachments would be included as an image only and that if the participant needed to access the original attachment, a link would be included at the bottom of each email. If a participant from Group 1 clicked a link to retrieve the original attachment, she was redirected to the warning page shown in Fig. 5. This initial version of the warning page provides two simple buttons to either proceed or abort.

Participants in Group 2 performed the same experiments as those in Group 1 where the only difference was the warning page that would open once a user requested an original attachment. Instead of the simple warning page used for Group 1, Group 2 uses a warning page that was inspired by Chrome's certificate warning page. In particular, the "Proceed" option was hidden, and would only become visible after the user clicked the "Advanced Options" link on the warning page (see Fig. 6).

Participants in Group 3 experienced the same warning page as those in Group 2. Additionally, these participants received a slightly different version of the introductory email which was designed to focus participants' attention on the pertinent content. To this end, we reduced the amount of text in the introductory email and highlighted operative words in bold-red typeface.

6) Results of User Study: After carrying out the experiment according to the protocols defined above and in Table I, we obtained the following results. The ten participants in the Control Group downloaded 25 (or 62%) of the 40 malicious attachments. Recall that each experiment features four malicious attachments, and thus ten participants will receive a total of 4 x 10 malicious attachments. This protocol was conducted to assess the baseline behavior of users when they receive email attacks without the protection of PellucidAttachment. Group 0 warned by a pop-up

TABLE II
DOWNLOAD BEHAVIOR PER PROTOCOL

	Benign Attachment		Malicious Attachment	
Downloaded	✓		✓	
Not Downloaded		✓		✓
Control	20 (100%)	0 (0%)	25 (62.5%)	15 (37.5%)
Group 0	20 (100%)	0 (0%)	25 (62.5%)	15 (37.5%)
Group 1	18 (90%)	2 (10%)	15 (37.5%)	25 (62.5%)
Group 2	16 (80%)	4 (20%)	9 (22.5%)	31 (77.5%)
Group 3	24 (60%)	16 (40%)	13 (16.3%)	67 (83.7%)

to verify the sender before downloading any attachment and they downloaded 25 (62%) of the malicious attachments. Groups 1, 2, and 3 received processed emails where the original attachments are replaced with images depicting their content. Group 1 had 10 participants, and they downloaded 15/40 (37.5%) of the malicious email attachments in total (see Table II). The more explicit warning page featured in the experiments for the ten participants in Group 2 further reduced the number of malicious attachments that were opened to merely 9/40 (or 22.5%). Finally, the improved introductory email led to only 13/80 (or 16.3%) of downloaded malicious attachments among all 20 participants in Group 3. That is, PellucidAttachment reduced the probability of downloading a malicious attachment from 62.5% to only 16.3% with improvement of 3.8x when compared with the control group.

7) Discussion of Experiment Results: The aim of the user study is to show that our proposed approach indeed helps users to make better security decisions. Besides the clear improvements introduced by PellucidAttachment, there was a trend of opening malicious files across all of the groups. Hence, the user study was designed to deceive the participants. The content of the emails and the names of the attachments were chosen to lure users into downloading the attachments. Forty participants opened at least one malicious attachment, and twenty-nine of those participants across all groups downloaded the first malicious file they received, independent of the group they were assigned to. However, once participants learned about the relationship between the attachments and their images embedded in the emails, they read emails with attachments more carefully, and the frequency of downloading malicious documents decreased. Although our study was deliberately designed to deceive users, using our system helped them avoid downloading malicious attachments and decreased the malicious attachment download rate from 62.5% to 16%. There is also a decrease in downloading benign attachments. PellucidAttachment already showing the preview image of the attached file. When PellucidAttachment is activated for Group 1, 2, and 3 the download rate decreased to 24/40 (60%). Therefore, users might not intend to download them on occasions. Users tend to download the original attachment when they need to modify or update or keep the original attached file.

To assess the utility of the different design decisions in PellucidAttachment we assess the following:

Null Hypothesis 1: The probability of downloading malicious attachments is independent of the assigned experimental group and is not effected by improvement protocol.

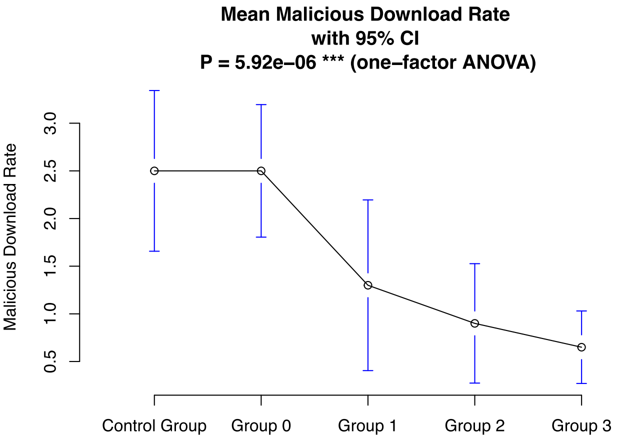


Fig. 8. One-factor ANOVA test results for significance between malicious attachment group download rates.

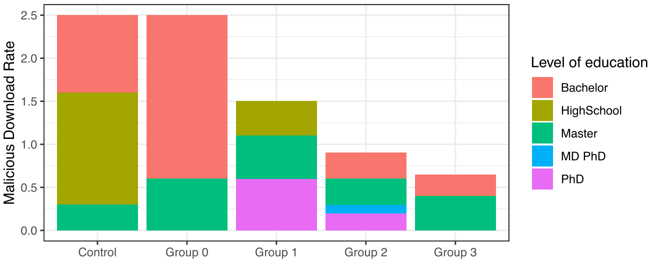


Fig. 9. Level of Education of participants and malicious attachment group download rates.

TABLE III
P-VALUES OF THE THREE PROTOCOLS AGAINST THE CONTROL GROUP

Groups	P-Value
Control Group vs. Group 0	1.0
Control Group vs. Group 1	0.1316
Control Group vs. Group 2	0.0046
Control Group vs. Group 3	0.0001

To test the null hypothesis, we performed one-factor ANOVA test for significance between malicious attachment group download rates. Fig. 8 shows the statistically significant decreasing trend in download rates ($P = 0.000187$). Level of education of participants in each group is shown in Fig. 9. Using Tukey test, we also tested the pairwise response differences between the control group and each of the four groups. As Table III illustrates, individuals using PellucidAttachment, especially in Group 2 and 3, had significantly lower malicious attachment downloading rates relative to the control group. Unfortunately, a pop-up warning page didn't help users because it did not have any information available for users to assess. We conclude that using an explicit introductory email and a warning page modeled after Chrome's certificate warning significantly reduce the probability that users open malicious attachments. In our experiments, users of PellucidAttachment were almost four times less likely to open malicious attachments than users from the control group. As such, we postulate that PellucidAttachment provides a significant increase in the security posture of regular email users against email-borne threats.

TABLE IV
SECURITY AWARENESS QUESTIONNAIRE SCORES BY GROUPS

	Password	Virus	Habit	Deletion	Attack	Confidence
# of questions	5	3	5	2	3	2
F-value	0.41	0.000	0.41	0.000	0.478	0.773
P-value (ANOVA)	0.81	1.000	0.801	1.000	0.752	0.551

8) Security Awareness Survey: At the end of each experiment, we asked each participant to complete questions about their security awareness. To this end, we administered a modified version of the Security Awareness Survey published by the SANS Institute [54]. We selected 20 out of 25 questions and grouped them into six sections.

The first section contained five questions about password habits of the participants. The second section contained three questions about anti-virus usage and virus experience. The third section contained five questions that covered computer security practices and habits of the participants such as whether they backup their data. The fourth section contained two questions concerning the participants' knowledge of data deletion. The fifth section had three questions that asked about online attacks, such as phishing. The sixth section had two questions which measured the security knowledge confidence of a participant. We collected the responses and compared the frequency of given response in each group.

Table IV summarizes the test statistics of responses to awareness questions based on six categories. Each statistic shows the statistical significance of difference between the responses from 4 groups. None of the question categories demonstrate significant difference indicating that individuals assigned to groups were uniformly distributed in terms of their security awareness backgrounds and leads us to assure an unbiased conclusion.

VIII. DISCUSSION AND LIMITATIONS

Spam and phishing prevention tools exist to keep users away from any malicious emails. In this study we focused on malicious email attachments, which might be in the form of spam, phishing, spear phishing, or even from a legitimate source where the sender is unaware of malicious activity contained inside the attached file.

Our user study aims to answer RQ2 and show the effectiveness of security warning design in the context of HCI. Although we demonstrated PellucidAttachment 's effectiveness, there is potential for further in-depth research in security warning design. The effectiveness of security warning designs in avoiding malicious email attachments can be explored by collecting more interaction information such as eye tracking [26], mouse tracking, and post-experiment user experience survey [27].

One observation we had during our user study is that there is a decrease in downloading benign attachments. Since PellucidAttachment is already showing the preview image of the attached file, users tend to download the original attachment when they need to modify or update or keep the original attached file. For future studies, this could be further investigated to lower the download rate to a minimum and benefit from local storage consumption.

Based on our thread model, while PellucidAttachment prevents the spreading of generic malicious documents that are crafted for a general audience, users will still be vulnerable to malicious email attachments that are specifically crafted to render to a meaningful image that convinces the user to acknowledge the security warning, download, and finally open the original file. However, PellucidAttachment significantly raises the bar for the attacker by requiring an image to be semantically meaningful to elicit this user behavior. When users are provided the preview of attachments, if the attacker does not put in the effort to create an email with a malicious attachment that has a meaningful preview, as it is validated in the user study, users are more likely to recognize and avoid downloading the attachment.

One of PellucidAttachment's limitations is the lack of support for non-visual file formats. PellucidAttachment cannot render files that have no meaningful visual representation (e.g., binary files, or compressed archives). If the attacker creates files that cannot be successfully converted to PNG images the user is left without any clue. However, as we can see through the statistics of not downloading the original files for both benign and malicious cases, after using PellucidAttachment for a while the user gets familiar with its capability and limitations. Moreover, according to VirusTotal statistics [55] -which has been used in 115 academic papers between 2008-2018 [56]-, PellucidAttachment covers the majority of the top 10 malicious file types that typically include malicious functionality.

Another factor that needs to be considered is the scalability of the practical implementation of our proposed approach. PDF conversion takes a considerably long time and uses a lot of memory, especially when the PDF has multiple pages. However, we can assume that this overhead occurs on the MTA before the user receives the email. Yet, for email providers, dynamically scanning a malicious attachment consumes more resources than what PellucidAttachment needs to prepare a preview of the attached files.

PellucidAttachment 's rendered images can be used as a data source for a machine learning model to provide users a new informative warning signal about the maliciousness of the email attachment. Moreover, currently available malicious email attachment techniques can be supported by additional information gained by PellucidAttachment. For example, certain cues of an email that are gathered through dynamic analysis can be combined with the rendered images to increase malicious email attachment detection rate.

PellucidAttachment 's email content filter only replaces email attachments with corresponding PNG image versions. Our approach protects users from malicious documents by blocking these artifacts before they are downloaded to the victim's system. Clearly, the modifications that PellucidAttachment performs on emails would break any cryptographic signatures. However, PellucidAttachment could be augmented to either pass through (unmodified) cryptographically signed emails from verified and trusted senders or to re-sign emails with a trusted identity. The threat model of cryptographic signatures states that attackers cannot forge valid signatures of verified senders, and thus such a mechanism would allow the small number of cryptographically signed emails to be authenticated correctly.

IX. Conclusion

In this paper, we proposed a novel defense mechanism against the prevalent threat of malicious email attachments. The core insight of our work is that today, email recipients have insufficient information to make an informed decision on whether a given attachment is benign (i.e., can be opened without concern) or malicious (i.e., opening the attachment poses a security risk). Our prototype implementation of PellucidAttachment narrows this information gap and replaces all attachments with images of their content. The conversion applied by PellucidAttachment strips any potentially malicious traits of an attachment while preserving the attachment's visual appearance. This methodology provides additional information to users and allows them to make better-informed decisions on how to handle email attachments.

We evaluated PellucidAttachment with an experiment on 39 malicious attachments that attack various vulnerabilities in real-world software. The transformations applied by PellucidAttachment successfully rendered all attacks ineffective. Additionally, we performed an extensive user study ($n = 60$) that measures and demonstrates the effectiveness of PellucidAttachment to protect potential victims from email-borne attacks. Our results indicate that PellucidAttachment reduces the probability for an untrained user to open a malicious email attachment by a factor of almost 4.

These results demonstrate that PellucidAttachment significantly raises the bar for attackers that seek to infect their victims through malicious email attachments.

References

[1] CVE-2020–681, 2022. [Online]. Available: https://www.cve.org/CVERecord?id=CVE-2020--6819

[2] CWE-416: Use-after-free, 2022. [Online]. Available: https://cwe.mitre.org/data/definitions/416.html

[3] Calyptix, DNC Hacks: How Spear Phishing Emails Were Used, 2016. [Online]. Available: http://www.calyptix.com/top-threats/dnc-hacks-how-spear-phishing-emails-were-used/

[4] S. Goel, K. Williams, and E. Dincelli, "Got phished? Internet security and human vulnerability," *J. Assoc. Inf. Syst.*, vol. 18, no. 1, 2017, Art. no. 2.

[5] D. Oliveira et al., "Dissecting spear phishing emails for older versus young adults: On the interplay of weapons of influence and life domains in predicting susceptibility to phishing," in *Proc. ACM CHI Conf. Hum. Factors Comput. Syst.*, ser. CHI '17. New York, NY, USA, 2017, pp. 6412–6424. [Online]. Available: http://doi.acm.org/10.1145/3025453.3025831

[6] S. Duman, K. Kalkan-Cakmakci, M. Egele, W. Robertson, and E. Kirda, "EmailProfiler: Spearphishing filtering with header and stylometric features of emails," in *Proc. IEEE 40th Annu. Comput. Softw. Appl. Conf.*, 2016, pp. 408–416.

[7] L. F. Cranor, "Can phishing be foiled?," *Sci. Amer.*, vol. 299, no. 6, pp. 104–110, 2008.

[8] TrendLabsSM, APT, "Spear-phishing email: Most favored apt attack bait," Trend Micro, 2012. [Online]. Available: http://www.trendmicro.com.au/cloud-content/us/pdfs/security-intelligence/white-papers/wp-spear-phishing-email-most-favored-apt-attack-bait.pdf

[9] E. M. Rudd, R. Harang, and J. Saxe, "MEADE: Towards a malicious email attachment detection engine," in *Proc. IEEE Int. Symp. Technol. Homeland Secur.*, 2018, pp. 1–7.

[10] R. Balzer, "Assuring the safety of opening email attachments," in *Proc. DARPA Inf. Survivability Conf. Expo. II*, 2001, pp. 257–262.

[11] S. Dong-Her, C. Hsiu-Sen, C. Chun-Yuan, and B. Lin, "Internet security: Malicious e-mails detection and protection," *Ind. Manage. Data Syst.*, vol. 104, no. 7, pp. 613–623, 2004.

[12] M. G. Schultz, E. Eskin, E. Zadok, M. Bhattacharyya, and S. Stolfo, "MEF: Malicious email filter: A UNIX mail filter that detects malicious windows executables," in *Proc. USENIX Annu. Tech. Conf. - FREENIX Track*, Boston, MA, USA, Jun. 2001.

[13] P. Laskov and N. Šrndić, "Static detection of malicious Javascript-bearing PDF documents," in *Proc. ACM 27th Annu. Comput. Secur. Appl. Conf.*, 2011, pp. 373–382.

[14] C. Smutz and A. Stavrou, "Malicious PDF detection using metadata and structural features," in *Proc. ACM 28th Annu. Comput. Secur. Appl. Conf.*, 2012, pp. 239–248.

[15] N. Šrndic and P. Laskov, "Detection of malicious PDF files based on hierarchical document structure," in *Proc. 20th Annu. Netw. Distrib. Syst. Secur. Symp.*, 2013, pp. 1–16.

[16] D. Maiorca, I. Corona, and G. Giacinto, "Looking at the bag is not enough to find the bomb: An evasion of structural methods for malicious PDF files detection," in *Proc. 8th ACM SIGSAC Symp. Inf. Comput. Commun. Secur.*, 2013, pp. 119–130.

[17] D. Liu, H. Wang, and A. Stavrou, "Detecting Malicious Javascript in PDF through Document Instrumentation," in *Proc. IEEE/IFIP 44th Annu. Int. Conf. Dependable Syst. Netw.*, 2014, pp. 100–111.

[18] I. Hopper, "Destructive 'I LOVE YOU' computer virus strikes worldwide," CNN Interactive Technol., 2000. [Online]. Available: https://edition.cnn.com/2000/TECH/computing/05/04/iloveyou.01/

[19] R. C. Dodge Jr., C. Carver, and A. J. Ferguson, "Phishing for user security awareness," *Comput. Secur.*, vol. 26, no. 1, pp. 73–80, 2007.

[20] FBI, spoofing and phishing, 2022. [Online]. Available: https://www.fbi.gov/how-we-can-help-you/safety-resources/scams-and-safety/common-scams-and-crimes/spoofing-and-phishing

[21] FBI, business email compromise, 2022. [Online]. Available: https://www.fbi.gov/how-we-can-help-you/safety-resources/scams-and-safety/common-scams-and-crimes/business-email-compromise

[22] M. Bhattacharyya, S. Hershkop, and E. Eskin, "MET: An experimental system for malicious email tracking," in *Proc. ACM Workshop New Secur. Paradigms*, New York, NY, USA, 2002, Art. no. 3.

[23] S. Stolfo, S. Hershkop, K. KeWang, and O. Nimeskern, "EMT/MET: Systems for modeling and detecting errant email," in *Proc. DARPA Inf. Survivability Conf. Expo.*, 2003, pp. 290–295.

[24] L. Muniandy, "Phishing: Educating the Internet users - a practical approach using email screen shots," *IOSR J. Res. Method Educ.*, vol. 2, no. 3, pp. 33–41, 2013.

[25] J. Petelka, Y. Zou, and F. Schaub, "Put your warning where your link is: Improving and evaluating email phishing warnings," in *Proc. ACM CHI Conf. Hum. Factors Comput. Syst.*, ser. CHI '19. New York, NY, USA, 2019, pp. 1–15. [Online]. Available: https://doi.org/10.1145/3290605.3300748

[26] L. Jaeger and A. Eckhardt, "Eyes wide open: The role of situational information security awareness for security-related behaviour," *Inf. Syst. J.*, vol. 31, no. 3, pp. 429–472, 2021. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1111/isj.12317

[27] M. Gutfleisch, M. Peiffer, S. Erk, and M. A. Sasse, "Microsoft office macro warnings:a design comedy of errors with tragic security consequences," in *Proc. ACM Eur. Symp. Usable Secur.*, ser. EuroUSEC '21. New York, NY, USA, 2021, pp. 9–22. [Online]. Available: https://doi.org/10.1145/3481357.3481512

[28] Deepmail, 2019. [Online]. Available: http://www.qualitia.co.jp/product/dm

[29] Cybermail, 2019. [Online]. Available: https://www.cybersolutions.co.jp/product/cybermail

[30] Gmail, 2019. [Online]. Available: https://www.google.com/gmail/

[31] Outlook.com - microsoft free personal email, 2019. [Online]. Available: https://outlook.live.com/owa/

[32] A. Moshchuk, T. Bragin, D. Deville, S. S. D. Gribble, and H. M. H. Levy, “SpyProxy : Execution-based detection of malicious web content,” 2007. [Online]. Available: http://dl.acm.org/citation.cfm?id=1362903.1362906

[33] M. Radhakrishnan and J. A. Solworth, “Quarantining untrusted entities: Dynamic sandboxing using leap,” in *Proc. Comput. Secur. Appl. Conf.*, 2007, pp. 211–220.

[34] CVE-2016–7193, 2019. [Online]. Available: https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2016--7193

[35] J. Dechaux, E. Filiol, and J.-P. Fizaine, “Office documents: New weapons of cyberwarfare,” Hack. Lu, 2010. [Online]. Available: http://2015.hack.lu/archive/2010/Filiol-Office-Documents-New-Weapons-of-Cyberwarfare-paper.pdf

[36] Microsoft malware protection center, 2019. [Online]. Available: https://www.microsoft.com/security/portal/enterprise/threatreports_july_2015.aspx

[37] CVE-2016–1009, 2019. [Online]. Available: https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2016--1009

[38] “Adobe reader out-of-bounds indexing remote code execution vulnerability,” 2019. [Online]. Available: http://www.zerodayinitiative.com/advisories/ZDI-16-191/

[39] C. Smutz and A. Stavrou, “When a tree falls: Using diversity in ensemble classifiers to identify evasion in malware detectors,” in *Proc. Netw. Distrib. Syst. Secur. Symp.*, 2016, pp. 659–673.

[40] Imagemagick, 2019. [Online]. Available: https://www.imagemagick.org

[41] CVE-2016–3714, 2019. [Online]. Available: http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2016--3714

[42] Libpng, 2019. [Online]. Available: http://www.libpng.org/pub/png/libpng.html

[43] CVE-2016–10087, 2019. [Online]. Available: https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2016--10087

[44] L. Caviglione et al., “Tight arms race: Overview of current malware threats and trends in their detection,” *IEEE Access*, vol. 9, pp. 5371–5396, 2021.

[45] D. Akhawe and A. P. Felt, “Alice in warningland: A large-scale field study of browser security warning effectiveness,” in *Proc. USENIX Secur. Symp.*, 2013, vol. 13, pp. 257–272.

[46] D. Modic and R. Anderson, “Reading this may harm your computer: The psychology of malware warnings,” *Comput. Hum. Behav.*, vol. 41, pp. 71–79, 2014.

[47] S. Egelman, L. F. Cranor, and J. Hong, “You’ve been warned: An empirical study of the effectiveness of web browser phishing warnings,” in *Proc. ACM SIGCHI Conf. Hum. Factors Comput. Syst.*, 2008, pp. 1065–1074.

[48] Postfix, 2019. [Online]. Available: http://www.postfix.org/

[49] Kernel virtual machine, 2017. http://www.linux-kvm.org/

[50] Libreoffice, 2019. [Online]. Available: https://www.libreoffice.org/

[51] Ghostscript, 2019. [Online]. Available: http://www.ghostscript.com/

[52] NGINX, 2019. [Online]. Available: https://nginx.org/

[53] Virustotal, 2019. [Online]. Available: https://www.virustotal.com/

[54] Security awareness survey, 2019. [Online]. Available: https://securingthehuman.sans.org/media/resources/business-justification/security-awareness-survey.pdf

[55] Virustotal statistics, 2019. [Online]. Available: https://www.virustotal.com/en/statistics/

[56] S. Zhu et al., “Measuring and modeling the label dynamics of online anti-malware engines,” in *Proc. 29th USENIX Conf. Secur. Symp.*, USA, 2020, pp. 2361–2378.