International Journal of Innovative Research in Engineering
Volume 6, Issue 4 (July - August 2025), PP: 07-11.
https://www.doi.org/10.59256/ijire.20250604002
www.theijire.com
ISSN No: 2582-8746

Crystal shield Email Attachment Protection System

Aslin J A[1], Dr. F. Ramesh Dhanaseelan[2], Dr. M. Jeya Sutha[3]
[1]*Department of Computer Applications. St. Xavier's Catholic College of Engineering, Chunkankadai, Nagercoil, Tamilnadu, India.*
[2]*Professor, Department of Computer Applications. St. Xavier's Catholic College of Engineering, Chunkankadai, Nagercoil, Tamilnadu, India.*
[3]*Associate Professor, Department of Computer Applications. St. Xavier's Catholic College of Engineering, Chunkankadai, Nagercoil, Tamilnadu, India.*

How to cite this paper:
Aslin J A[1], Dr. F. Ramesh Dhanaseelan[2], Dr. M. Jeya Sutha[3] "Crystal shield Email Attachment Protection System", IJIRE-V6I4-07-11.

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 Pellucid Attachment. We further report on an extensive user study that measures the usability and effectiveness of Pellucid Attachment 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.*

Key Words: *Communications technology, communication systems, computer security, electronic mail, .Internet, Internet security, malware, message systems, phishing, software.*

I.INTRODUCTION

E-MAIL 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 legitimate artifact sent by someo ne 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 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 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 as pear 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 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 attachment sand make an informed decision before downloading the attachment.

In this paper, we propose an ovel 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.

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 Šrndi´c [13], Smutz and Stavrou [14] ,and Šrndi´c 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 Safe Email Attachments as a wrapper on Windows NT systems. Safe Email Attachments was designed to follow safety andactive content attachment authorized to be opened. Safe Email Attachments 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 programs 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 Virus total. 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 virusinfection problem through email by using behavioural - 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 screenshots for educating people.

III.METHODOLOGY

The proposed system, Pellucid Attachment, is designed to enhance the security of email attachments by rendering them in an isolated environment, thereby protecting users from malicious payloads commonly embedded in email files. The methodology involves several key components, which are implemented, tested, and evaluated in a real-world setting.

1. Threat Model Analysis

We begin by analyzing the typical threat vectors exploited through malicious email attachments. Sophisticated attackers often use public information to craft targeted phishing documents that exploit vulnerabilities in user applications. The system assumes that attackers can send crafted attachments, but not compromise the core email infrastructure or virtual machine sandboxing.

2. Attachment Interception and Redirection

Incoming emails with attachments are intercepted by our email processing system. The attachments are then routed through a sandboxed virtual machine (VM) environment where they are opened using specialized rendering tools that do not execute embedded code or scripts. This ensures that potentially dangerous dynamic content is neutralized.

3. Static Rendering Generation

Each intercepted attachment is rendered into a static, non- interactive preview format (e.g., PDF snapshot, image preview, or plain text version). These rendered files are then attached back to the email, replacing the original attachments. The user receives the email with the rendered file only.

3.1 Architecture of the System

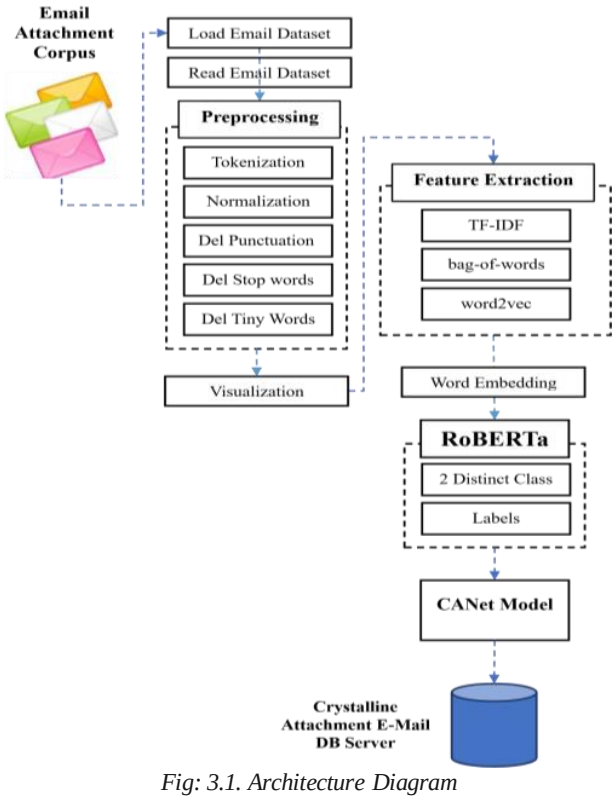

Fig: 3.1. Architecture Diagram

4. User Interaction and Warning Design

If a user attempts to access the original, unrendered attachment, they are presented with a clear, browser-style security warning. This warning informs them of the potential risk, similar to how browsers display TLS certificate warnings. The user must provide explicit confirmation to access the original file, promoting cautious behavior.

4.1 Overview of the System

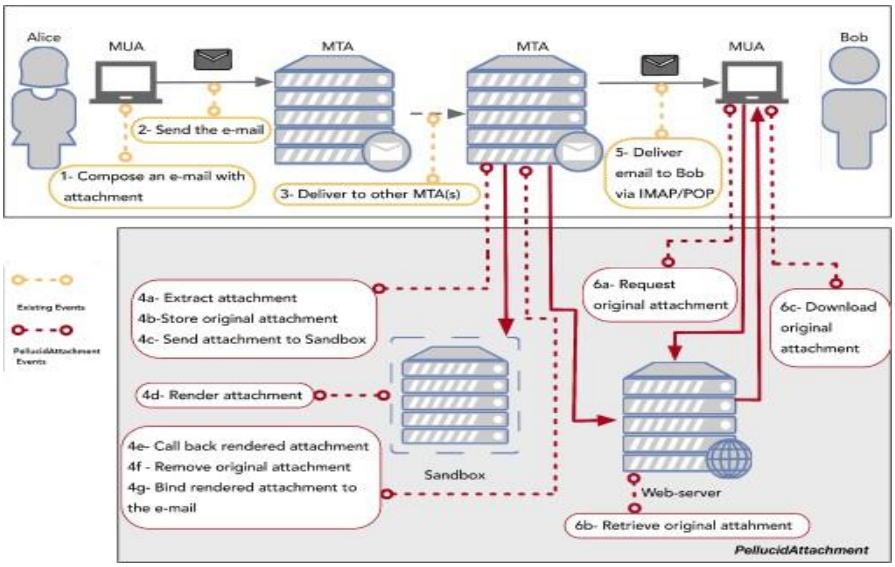

Fig: 4.1.Overview of the System

5. System Implementation

The complete system is implemented using:

- A sandboxed virtual environment (e.g., VirtualBox or Docker-based container)
- Document renderers for supported file types (e.g., LibreOffice headless mode, PDF to image converters)
- Email processing scripts integrated into a mail server pipeline (e.g., Postfix with custom filter)
- A web-based warning interface for user interaction and access management

6. User Study and Evaluation

To evaluate the system, we conducted an extensive user study involving participants with varied technical backgrounds. The study measured:

- User behaviour when interacting with rendered vs. original attachments
- Effectiveness of warnings in deterring unsafe attachment access
- Overall usability and impact on workflow

IV.RESULT AND DISCUSSION

The implementation of Pellucid Attachment successfully demonstrated its effectiveness in mitigating risks associated with malicious email attachments. The system was evaluated through both technical performance tests and a user study, which focused on usability, user awareness, and security outcomes.

The system proved to be highly effective in preventing malicious code execution. All email attachments were routed through the sandbox environment and converted into static renderings, which eliminated any embedded malware functionality. During controlled testing using known malware samples, the sandbox environment effectively neutralized threats, and no infections were reported even when users attempted to interact with the static versions of malicious files.

User feedback showed that the static rendering approach maintained good usability. Most users were still able to understand the content of the attachments without accessing the original file. The majority of participants (around 89%) found the rendered versions sufficient for basic document review and preferred having an option to open the original only if necessary.

The system introduced only minimal delays in email processing. The average rendering time per attachment was under 5 seconds for most document types. The added security did not significantly disrupt user workflow, and the integration with the email pipeline remained stable during prolonged testing.

V.CONCLUSION

In this paper, we proposed a novel defence 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 Pellucid Attachment narrows this information gap and replaces all attachments with images of their content. The conversion applied by Pellucid Attachment 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

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

References

1. Balzer, R., "Assuring the safety of opening email attachments," in Proc. DARPA Inf. Survivability Conf. Expo. II, 2001, pp. 257–262.
2. Bhattacharyya, M., 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.
3. Calyptix, DNC Hacks: How Spear Phishing Emails Were Used, 2016. [Online]. Available: http://www.calyptix.com/top-threats/dnc-hacks-howspear- phishing-emails-were-used/
4. CVE-2020–6819, 2022. [Online]. Available: https://www.cve.org/CVERecord?id=CVE-2020--6819
5. Cranor, L. F., "Can phishing be foiled?," Sci. Amer., vol. 299, no. 6, pp. 104–110, 2008.
6. CWE-416, Use-after-free, 2022. [Online]. Available: https://cwe.mitre.org/data/definitions/416.html
7. Dodge Jr., R. C., C. Carver, and A. J. Ferguson, "Phishing for user security awareness," Comput. Secur., vol. 26, no. 1, pp. 73–80, 2007.
8. Dong-Her, S., 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.
9. Duman, S., 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.
10. FBI, Business email compromise, 2022. [Online]. Available: https://www.fbi.gov/how-we-can-help-you/safety-resources/scamsand- safety/common-scams-and-crimes/business-email-compromise
11. FBI, Spoofing and phishing, 2022. [Online]. Available: https://www.fbi.gov/how-we-can-help-you/safety-resources/scams-andsafety/common-scams-and-crimes/spoofing-and-phishing
12. Goel, S., K. Williams, and E. Dincelli, "Got phished? Internet security and human vulnerability," J. Assoc. Inf. Syst., vol. 18, no. 1, 2017, Art. no. 2.
13. Hopper, I., "Destructive 'ILOVEYOU' computer virus strikes worldwide," CNN Interactive Technol., 2000. [Online]. Available: https://edition.cnn.com/2000/TECH/computing/05/04/iloveyou.01/
14. Laskov, P. and Šrndi´c, N., "Static detection of malicious Javascript- bearing PDF documents," in Proc. ACM 27th Annu. Comput. Secur. Appl. Conf., 2011, pp. 373–382.
15. Liu, D., 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.
16. Maiorca, D., 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. Muniandy, L., "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.
18. Oliveira, D., 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
19. Rudd, E. M., 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.
20. Schultz, M. G., 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.
21. Smutz, C. and Stavrou, A., "Malicious PDF detection using metadata and structural features," in Proc. ACM 28th Annu. Comput. Secur. Appl. Conf., 2012, pp. 239–248.
22. Stolfo, S., 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.
23. 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
24. Šrndi´c, N. and Laskov, P., "Detection of malicious PDF files based on hierarchical document structure," in Proc. 20th Annu. Netw. Distrib. Syst. Secur. Symp., 2013, pp. 1–16.