ENS 491-492 – Graduation Project

Final Report

Project Title

Smart Phish Generator

Group Members

Mete Harun Akçay

Emirhan Delican

Ataollah Hosseinzadeh Fard

Supervisor

Orçun Çetin

Date: 07.01.2024

1. EXECUTIVE SUMMARY

The IT security industry is bracing itself for an onslaught of new wave hacking styles which are powered by smart technology, machine learning, natural language processing and artificial intelligence. The emails, messages or news generated by utilizing such methods do not only trick individuals into disclosing their personal information, but they also pose a threat to companies and even governments, leading to tarnished reputations and billions of financial losses each year. The increase in phishing attacks recently shows the incompetence of existing anti-phishing tools; hence, points out the need for more efficient tools. This project, thus, focused on the implementation of a Smart Phish Generator, an AI-powered model that can generate highly convincing phishing emails. With the implementation of such model, it was aimed to contribute to the effectiveness of existing anti-phishing tools by discovering the vulnerabilities of them. Moreover, it was also aimed to raise awareness among humans about the evolving cyber threats. Sample emails collected by contributors of the project were parsed into small components as logo, image and texts. Google Images, Dall-E and OCR were used to produce new logos and images, whereas OpenAI's ChatGPT was used to generate a text related to given topic. Additionally, news collected from RSS feeds were tried to be converted into phishing emails. Initial results show that model was achieved to generate phishing emails from more than 90% of the inputs, 70% of which are determined to seem authentic, attractive and persuasive that can be identified as having the potential to deceive individuals in phishing attacks.

2. PROBLEM STATEMENT

Phishing can be defined as “a fraudulent activity that involves the creation of a replica of an existing web page to fool a user into submitting personal, financial, or password data”.[1] These fraudulent activities are becoming increasingly common in today's digital world, with cybercriminals using a variety of tactics to trick people into divulging their personal information or carrying out malicious activities.[2] Phishing attacks can take many forms, such as email, text messages, or social media messages, and can be highly targeted or more widespread in nature.

[1] Merwe, A. v. d., Marianne, L., and Marek, D. (2005). “Characteristics and responsibilities involved in a Phishing attack, in WISICT ’05: proceedings of the 4th international symposium on information and communication technologies. Trinity College Dublin, 249

[2] Ibid, 250

They can also be highly effective, with many people falling victim to these frauds every day. According to a study[3] on phishing attacks, the number of detected attacks has shown a significant upward trend over the past five years. Specifically, the study found that the number of unique phishing emails detected from 2017 to 2019 has increased by 150%. Similarly, there is almost 50% increase in the number of brands targeted by phishing attacks, from 2018 to 2019.

As a result of data breaches and sensitive information exposure due to successful phishing attacks, significant financial losses totaling billions of dollars occur. In fact, the report showing the total loss of 12 billion USD in 2018 estimates that the yearly damage will exceed 20 billion USD this year due to the increase in the phishing emails.[4] This increase can be explained by the new techniques developed lately. Advancement in Artificial Intelligence, specifically in Large Language Models (LLMs) pawed the way for cybercriminals to create more persuasive email contents and convey them to wide range of people, that it became more difficult to distinguish them from non-phishing content by experts.[5] Even the advanced anti-phishing tools were reported to be unable to detect "persuasive" phishing emails.[6]

Various papers focused on the aspects affecting the persuasiveness of an email were studied. It was found that emails having loss or reward-based influence techniques are more likely to be clicked by readers.[7] Specifically, a loss-based email (i.e., an email suggesting that the recipient will lose something if they fail to respond) was found to be more persuasive than a reward-based email (i.e., an email offering some reward for responding). Additionally, it is observed that emails having logos and footers were considered as more trustworthy than those without them.[8] Moreover, it is suggested that emails are more persuasive when the arguments directing

[3] APWG (2020). APWG phishing attack trends reports. 2020 anti-phishing work. Group, Inc. Available at: https://apwg.org/trendsreports/

[4] Karim, A., Azam, S., Shanmugam, B., Kannoorpatti, K., & Alazab, M. (2019). A Comprehensive Survey for Intelligent Spam Email Detection. *IEEE Access, 7*, 168261.

[5] Hazell, J. (2023). Large Language Models Can Be Used To Effectively Scale Spear Phishing Campaigns. p:2

[6] Roy, S., Naragam, K., & Nilizadeh, S. (2023). Generating Phishing Attacks using ChatGPT.

[7] Emma J. Williams & Danielle Polage (2019) How persuasive is phishing email? The role of authentic design, influence and current events in email judgements, Behaviour & Information Technology, 38:2, 192

[8] Ibid, 193

the reader to do something (e.g., click a button) are placed at the beginning or the end of the message.[9]

Considering the findings discussed above, this project aimed to develop a "Smart Phish Generator", an AI-powered automated system that converts daily emails to more persuasive phishing emails by changing its components from logo and images to texts and buttons. In addition to emails, another tool which scrapes news from various news agencies and converts them to phishing emails was also developed. Primary purpose of developing such model is to contribute to the effectiveness of existing anti-phishing tools by discovering the vulnerabilities of them by generating highly convincible phishing emails. Moreover, it is also aimed to raise awareness among email users about the evolving cyber threats.

2.1. Objectives / Tasks

2.1.1. Research on Phishing Emails, AI-based Phishing and Anti-phishing

Intended result: Have a better understanding of the nature of phishing emails, persuasive aspects of messages, Large Language Models (LLMs), anti-phishing tools.

2.1.2. Data Collection

Intended result: Collect a large number of real-world emails from various sources to observe the similarities, common components of persuasive emails.

2.1.3. Database Creation

Intended result: Store the emails that seem persuasive enough to be used in phish generation. Create a simple content merger to test the data, keep it as non-AI email generator.

2.1.4. Selecting and Optimizing the LLM

Intended result: Find the LLM with the most accurate outputs by testing various prompts. Determine the prompt to generate text, logo and image.

2.1.5. Automating the Program

Intended result: Create a program capable of autonomous operation, converting authentic emails to phish emails randomly without human intervention.

---

[9] Kim, D., & Hyun Kim, J. (2013). Understanding persuasive elements in phishing e-mails: A categorical content and semantic network analysis. *Online Information Review, 37*(6), 837

2.1.6. Testing the Model

Intended result: Test the persuasiveness of the outputs of the model by conducting a survey on participants, tasking them with distinguishing between authentic and phishing emails.

2.1.7. Create Project Documentation

Intended result: Prepare final report and presentation to publish the results of the project, hopefully contribute to the development of anti-phishing tools.

2.2. Realistic Constraints

2.2.1. Economic

There were several economic challenges of this project. OpenAI's API was used to generate email content. While emails having AI-generated images and logos cost approximately $0.045 in average, emails that only have texts as AI-generated components were observed to cost $0.005. Additionally, if it is decided to use the end-product with automation and serve continuously, there will also be a remote server cost which is estimated to be at most $10.

2.2.2. Environmental

Although this project is a software-based project, it is still worth considering any potential environmental impacts that may arise from it. Energy consumption and waste production associated with the use of computer equipment should be minimized in software projects. However, such problems were not encountered during the project due to the relatively small scale of the project.

2.2.3. Manufacturability

In order for this project to be useful, it must be able to utilize external software tools. This requires consideration of factors such as the availability of external AI models and accessibility of remote components planned to be used. Thus, in regard of this project it is crucial that OpenAI continuously serves its products as both the development phase and the final product depends on OpenAI services. On the other hand, this project did not require any mass production, or more generally, any hardware related productions.

2.2.4. Sustainability

As in all software-based projects, sustainability was one of the major constraints of this project. The developed software should be easily updatable and maintainable. Moreover, the technological and security practices used in the project may decide to be changed over time, in that case the new practices should be adoptable. The project was conducted considering these aspects so that the updates will be able to be applied without problems. Moreover, several ethical and legal issues that may arise in the lifetime of the project were taken into account to ensure the sustainability in the long term.

2.2.5. Security

Since the early phases of the project required personal email collection, special care was taken to ensure the confidentiality of data. It was made sure not to let data leaks, and use the data only in the scope of this project.

2.2.6. Copyright

The collected emails contained copyrighted materials such as logos and brand names, which may cause legal problems due to the nature of this project. Removing these materials was not considered to be a good idea since they were the key features of persuasive emails. Thus, various techniques to avoid copyright infringement without decreasing the persuasiveness of emails were used, which will be discussed in detail in later sections.

3. METHODOLOGY

Throughout the project, various methods/techniques were used from data collection to testing the results. Email collection was done by storing team members' emails received from various sources in a newly created email account. It was taken care to include emails that are pleasing to the eye, i.e. emails having decent structure with possibly visual components. In order to avoid having similar outputs, it was determined not to include multiple emails from the same source. On the other hand, news collection was done by gathering news from various RSS feeds. To have a variety of news, both global and local RSS feeds were used.

Detection of the components in the given email was one of the key aspects of the project. Initially, a program having unique functions to detect and label each component was developed utilizing Python's Beautiful Soup library as a html parser framework. While this

program was highly accurate in detecting logos and other images, it did not achieve well in detecting text components initially. Throughout the project, this labeler continuously improved as it is the foundation of every other process in this project. Those improvements were done by examining the structure of each email separately and extracting rules that can be applied to both in general and individual emails. For instance, if a specific image could not be detected as it could not be verified by the current ruleset, a new rule was only created for this image if it was also applicable to other emails in the dataset. This was the approach that was being followed to improve the labeling tool. Furthermore, when each text component (header, main text, button text, footer) was processed separately, output emails were lacking harmony among these components, consequently lowering the persuasiveness of the emails. Therefore, a new approach was developed for both detecting the text components and processing them.

Utilizing from the knowledge that was gained from analyzing the components of the emails, an algorithm that extracts the block of texts was designed. In this context, a block refers to the html tags that are highly likely to contain a text inside of it. For instance, an "a" tag and a "p" tag are likely to contain a text inside them. Although, it may seem an easy task at first, the challenge was to detect those blocks as a whole when the tags that contain texts were nested. For instance, "a" tags could have been detected easily; however, they sometimes do not mean anything alone, they are meaningful within the general block they belong to. The hardness of the problem was not coming from finding the unique blocks. Instead, it was coming from determining the scope of those blocks. In other words, the problem was to determine when to stop while tracing backwards to find the smallest block that contains something meaningful as a whole. The developed algorithm accomplishes this task with a high percentage. In the end, what it gives is the unique blocks that contain meaningful text constructed with nested text components. Having a list of unique blocks that together contains all of the texts inside of an email, everything was there to feed the AI model that was going to be used. At this point, it may be thought whether it was possible to give the entire html to the AI and ask it to change the content and the context. Having tried such a procedure, it was concluded that it does not work as expected as the input size becomes too large, and most of the part of an html is irrelevant, which diminishes the overall output quality of LLM, specifically "gpt-3.5-turbo-1106" model of OpenAI.

Different methods for converting and regenerating the components of emails and news were used. When the input was an email, while text generation was mainly done with the

help of LLMs, image and logo generation required different techniques. As discussed in the literature review, emails having logos are considered much more convincing than the ones without logos. However, keeping the original logos in the phishing emails may have caused copyright issues. To avoid such issues, an algorithm that reads the given logo with OCR, determines the borders of the letters, then replaces or removes selected letters was developed. By doing this, it was aimed to avoid copyrights without changing the overall structure of the logo so that readers do not notice the difference. Some examples of this method can be seen in figures below.

Figure 1. Original logo of IMDB

Figure 2. Altered logo of IMDB

Figure 3. Original logo of YouTube

Figure 4. Altered logo of YouTube

While this method was successful for avoiding copyrights, another approach was used in order to create completely new logos related to the given subject. In order to do this, Dall-E 3, an AI model that creates images from a description in natural language was used. The motivation behind using Dall-E was that the sender company of an email had to be related with the subject of the email. For instance, an email promoting a discount on shoes is expected to be sent from a shoe store, not necessarily the company that is the sender of the original email. Therefore, the program was designed to use Dall-E to create a logo for a company promoting the given subject as input. By doing so, it was aimed to increase the persuasiveness level of the email. Two examples of AI-created logos along with their input

prompts are shown below. It is worth noting that not all of the results were as promising as these examples; therefore, all three of the approaches were used interchangeably during the project.

Figure 5. Dall-E's logo with the prompt: "Create a logo for a company promoting discounts on shoes".

Figure 6. Dall-E's logo with the prompt: "Create a logo for a company promoting new hairstyling techniques".

Main image of an email is one of the most important components since it serves as the initial visual impression, taking the reader's attention and making them more likely to engage with the message. Therefore, it was aimed to find attractive images for output emails. However, being appealing to the reader was not the only necessity, it also had to be related to the content of the email since it would cast doubt on the idea that the email was a phishing email otherwise. Similar to changing the logo, different methods for setting the image were implemented. When the sample emails were stored in the database and the similarities were analyzed, it was realized that most of the images of promotional emails were similar in terms of their style which can be referred as "vector art", or "illustration art". Thus, first approach to find a proper image for an email was to query Google Images to get an image of illustration of the given subject. For instance, if the subject was "free coupons", it was expected from the model to generate an email stating that the reader has earned free coupons and has to click the button to receive them. For this email, the model searched "free coupons illustration" on Google Images and replaces the image with the original image in the input email. Two instances for this approach can be seen below.

Figure 7. Free Coupon vector art

Figure 8. Student Discounts vector art

Even though the results were usually decent, it was not always the case, unfortunately. Thus, a similar method that was used in logo generation was tried. Depending on the subject, Dall-E 3 was queried to generate an attractive image. Two instances of Dall-E's outputs can be seen below, when the queries were "exciting news in Artificial Intelligence" and "amazing discounts", respectively. These generated images were stored in free tier service of firestore storage for both global access and further use. This was necessary as OpenAI deletes the generated images from its storage after one hour from generation.

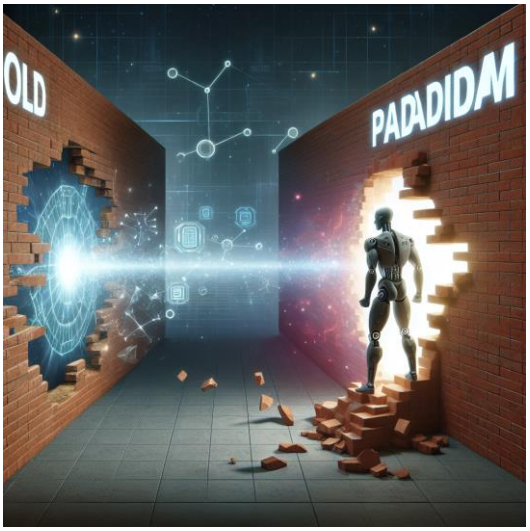

Figure 9. Dall-E's image with the subject: "Exciting News in Artificial Intelligence"

Figure 10. Dall-E's image with the subject: "Amazing Discounts"

Similar to the logo generation step, both of the discussed methods were used interchangeably since neither one was determined to be superior to the other.

Headers, regular text components, and buttons together basically create the context of the email. Thus, in order to create a new context, they have to be changed accordingly. However, changing each of them separately would not work, as they mean something as a whole. Therefore, they have to be changed in the same manner to preserve the integrity of the general email. Preserving this connectivity of those identified components plays a very crucial role for the email being convincing at the end. If the integrity and the harmony between those components cannot be preserved, the end product becomes useless as it would be far away from the real emails that big companies send every day. This has been taken into account very seriously while designing, implementing and evaluating the product. Furthermore, as mentioned previously, the labeling tools were continuously improved, especially in terms of detecting text components. Using the latest version of this labeler, the smallest unit of meaningful text blocks could be caught with a decent precision. Collecting each unique text block within an email together with their html tags, it was managed to change them around a new context while preserving both the integrity and the complex structure of the nested tags. Although the "button" may seem under the wrong sub-topic, it is indeed not. The buttons are no different than texts or headers in this context, as they are constructed with text components, and they mean something alone. Thus, changing a button is no different than changing regular paragraphs or titles. The content change for the caught components is being done by OpenAI's "gpt-3.5-turbo-1106" model. The caught components are appended into the prompt, and in the output, it was expected to observe the changed version of the appended tags. Although the gpt-4 achieves slightly better, it is too costly for such large inputs and outputs. Therefore, the model was optimized to work well with the previously mentioned gpt-3.5 model. After the content change is completed and returned back, the received components were put back to where they belong in the html.

On the other hand, converting current news to phishing emails was done with different methods. First step was the news collection phase. This step was done by utilizing RSS feeds of globally well-known news agencies such as 'CNN', 'BBC', 'Reuters', and some local news such as 'Sözcü', 'Cumhuriyet', 'Hürriyet', and more. In addition, to increase the variation and reduce the bias, some other local news agencies from other countries, such as 'Ukrinform' from Ukraine, were used. All the gathered news are stored in a JSON file, which later can be used by the phish generator tool as many times as possible. Figure given below is a sample view of that collected news.

```
{
    "result": 609,
    "date": "2023-12-22T10:51:18.560942",
    "data": [
        {
            "title": "10 günde 6 motosiklet çaldı! Kıskıvrak yakalandı",
            "date": "Güncelleme Tarihi: Aralık 22, 2023 10:42",
            "source": "https://www.hurriyet.com.tr/gundem/10-gunde-6-motosiklet-caldi-kiskivrak-yakalandi-42379972",
            "content": "İl Emniyet Müdürlüğü Asayiş Şube Müdürlüğü, Oto Hırsızlık Büro Amirliği ekipleri, Yüreğir ilçesinde 10 günd
        },
        {
            "title": "US court revives Nirvana album cover lawsuit",
            "date": "Published On 22 Dec 2023",
            "source": "https://www.aljazeera.com/?t=1703226118",
            "content": "A United States court has revived a lawsuit accusing the rock band Nirvana of publishing child pornography
            "images": []
        },
        {
            "title": "EYP yaptı, polisi görünce 'intihar edecektim' dedi",
            "date": "Güncelleme Tarihi: Aralık 22, 2023 10:24",
            "source": "https://www.hurriyet.com.tr/gundem/eyp-yapti-polisi-gorunce-intihar-edecektim-dedi-42379963",
            "content": "Olay, 20 Aralık'ta saat 11.00 sıralarında, Seyhan ilçesi Kayalıbağ Mahallesi'nde meydana geldi. Psikolojik
        },
        {
            "title": "Deniz'in katillerine İstinaf'tan kötü haber",
            "date": "\nYayınlanma: 10:45 - 22 Aralık 2023\n",
            "source": "https://www.sozcu.com.tr/deniz-in-katillerine-istinaf-tan-kotu-haber-p10256",
            "content": "Diyarbakır'da bir eğlence merkezinde çalışan Deniz Ketir (27), iki yıl önce babası Celal'in azmettirmesiyle
        },
```


Figure 11. Four instances of collected news

After the news collection, the content generation was the main focus. The content contains title, logo, body image, body text, and sender company. At the beginning, all parts except logo and body image were generated using OpenAI's "gpt-3.5-turbo-1106". Among all the news, one of the news was randomly selected and passed to the AI model as input, and the model was expected to return title, text, button text, and company name. The logo was being found by searching for the company name. Moreover, the body image was being found by searching for the title. However, it became evident that in some cases the AI model was giving imaginary company names and as a result it was impossible to find logos by searching them in google. Figure given below is an example of such cases when generation tool failed to find a valid logo.

Figure 12. Logo selection for Palestine Red Crescent Society

As further improvements were being done, static HTML templates that try to mimic email from 19 different news agencies were created. These news agencies were selected from some of the rss feed sources, which are Anadolu Ajansı (AA), İhlas Haber Ajansı (İHA), Demirören Haber Ajansı (DHA), Habertürk, NTV News, Hurriyet, TRT Haber, TRT

World, Sözcü, Reuters, Bloomberg, BBC, CNN, Al Jazeera News, The New York Times, The Washington Post, The Guardian, The Wall Street Journal, and Ukrinform. Following is one of the static templates.

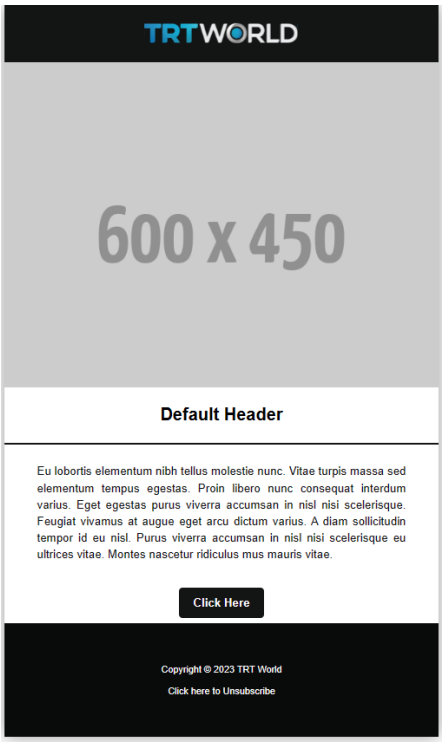

Figure 13. Static template instance

As a result, with these new static templates only title, text, and body image were needed. In addition, AI model's prompt modified in a way that it generates search queries to find body images in google. Hence, this new approach is able to find more related images in google. At the beginning, the generated news-based emails were more flexible in terms of theme since the AI model was determining sender company, but in this last version the sender company is selected randomly among predefined news agencies.

Having developed the phish generator, some techniques to use the model without human intervention were applied. An automation tool was developed to perform a phishing conversion to the incoming emails. This automation tool is directly connected to the shared email account, and whenever a new email arrives to the inbox, the tool immediately downloads the html version of this email and passes it to the phish generator tool. To perform such an operation, the topic also should be selected automatically. For this reason, a hardcoded list of topics were embedded into the code, and the selection was done

randomly from the hardcoded list during the runtime. In short, the automation tool acts as an oracle between email account and phish generator, in which the input is received email, and the output is its phished version.

4. RESULTS & DISCUSSION

The final product of the project was aimed to be a Smart Phish Generator, an AI-powered model that converts daily, authentic emails to persuasive phishing emails. Fortunately, it can be said that this aim was realized; hence, it is successfully completed. The results of the project in more detail are discussed in this section.

Initially, the topics for generating a phishing content were given as inputs. To make the system completely automated, a dictionary containing several topics were created. It included 10 loss-based topics (e.g. ‘login from unknown device’), 8 reward-based topics (e.g. ‘free coupons received’) and 7 neutral topics (e.g. ‘popular movies of the week’). As already mentioned, team members have collected 100 authentic emails to use them in the scope of this project. The final product was designed such that first it selects an email from the email inbox randomly, then it selects a topic from the dictionary randomly, finally it converts the selected email to a phishing email related to the selected topic. This means that there are 25 x 100 = 2500 possible outputs that the model may produce. In fact, this number is even more since the LLM can create different outputs with same inputs. Also, since the model was not designed in a static manner, it is always possible to add new emails and topics to widen the range of outputs of the model without affecting its performance.

Having explained the potential number of outputs of the model, it is worth mentioning that being able to generate high number of outputs does not necessarily mean that the model is successful. To observe the model’s success, each email was put into model with a randomly selected topic. Initially, the model was investigated to observe whether it failed to convert some emails. It was observed that model was failed to generate 8 of the 100 emails in the first place. These failures were mainly stemming from the insufficient capability of the LLM model. This insufficiency was more obvious especially when relatively long and complex inputs were given to the model. For instance, the detected tags by the labeler were appended into the prompt in a structured way, and in the output, it was expected to get every one of these tags with their content being modified in the same order. However, in some of the emails it was observed that some of the tags that were in the input

did not appear in the output which breaks down the assumption that was made. Using a more capable model, such as gpt-4, pretty much solves the problem, but increases the cost nearly ten times.

Secondly, the emails that model successfully generated were visually examined to see whether they achieved sufficiency in terms of their level of persuasiveness. This analysis was done by observing the overall structure, colors, logos and images of the emails. Since the structure of input emails was preserved, there were no problems related to this. However, main images were observed to be entirely relevant to the rest of the email at times. Specifically, loss-based emails with images usually seemed 'lacking in seriousness'. Hence, it was decided that the model would use email samples without images to generate most of the loss-based emails. However, as it can be seen below, some the loss-based emails with images that seemed alright made it to the next step of testing phase.

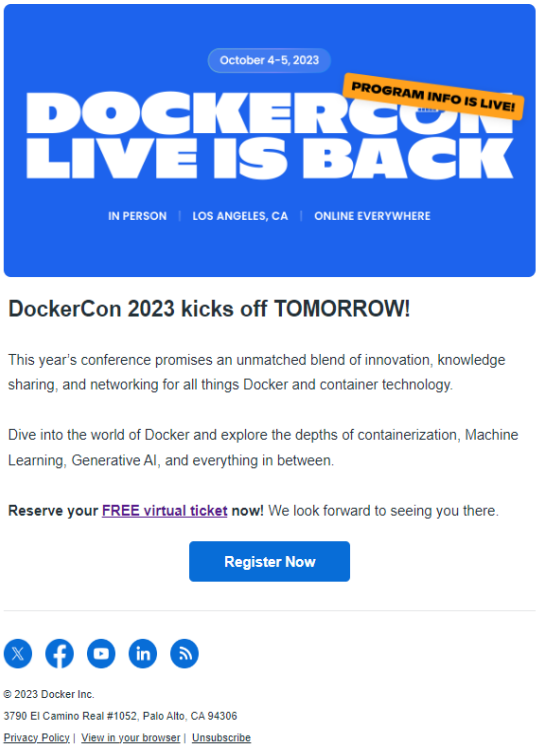

Figure 14. Original email sent by Docker

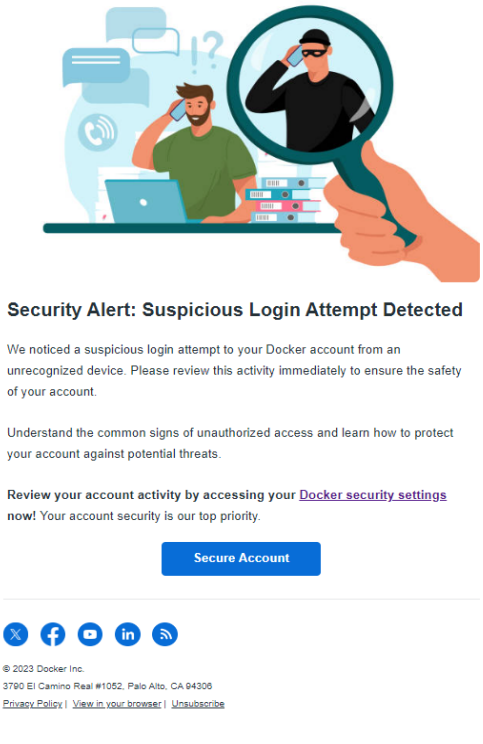

Figure 15. Output email when the topic was 'Suspicious Login Attempt'

Nonetheless, most of the loss-based emails were generated without images, as can be seen in figures below.

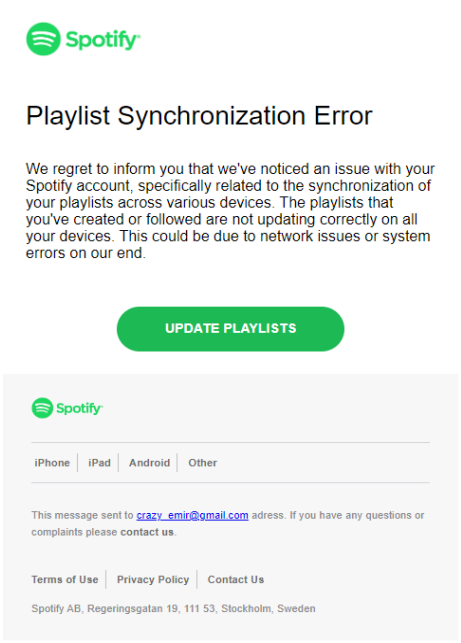

Figure 16. Original email sent by Spotify

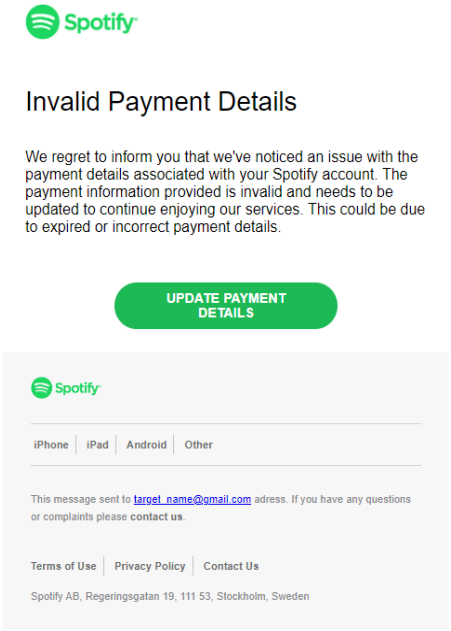

Figure 17. Output email when the topic was 'Unsuccessful Payment'

On the other hand, images in reward-based emails were considered as boosters, i.e. they seemed to enhance the effect of the messages in the emails. For instance, an email promoting summer sales with an illustration of beach, sun etc. seemed more attractive than the ones without images. Likewise, as it can be seen below, an illustration of a gift box containing crypto coins in a crypto-related email is more likely to attract the user to click the button to receive his/her gift.

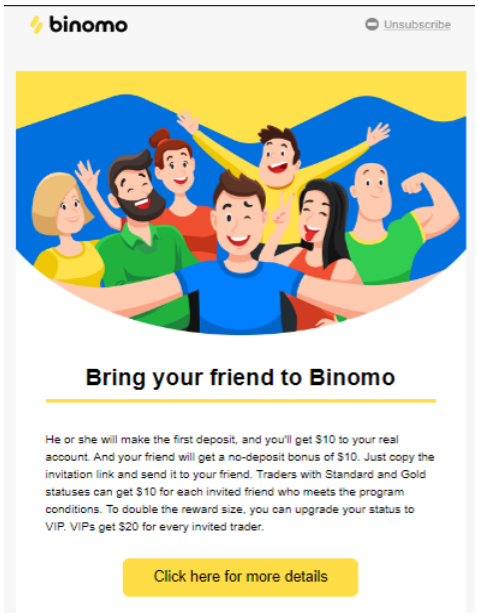

Figure 18. Original email sent by Spotify

Figure 19. Output email when the topic was 'Crypto Gift'

Finally, the generated texts for output emails were analyzed in terms of their conciseness, coherence, correctness and persuasiveness. None of the emails were observed to have grammatical errors. Likewise, none of the emails had unnecessary words or repetitions. However, conciseness of some of the emails were questionable. The texts in such emails did not flow logically, and the different ideas were observed not to be connected in a coherent manner. Overall, they did not seem to be human written. Thus, 21 emails were eliminated at this step. The remaining 71 emails were classified as 'persuasive emails'.

Initial observations of the outputs of news-based phishing emails were promising either. Figures given below are samples from dynamic and static templates, respectively. Although, dynamic template seems more convincing and realistic, the ratio of dynamically well generated emails over total generations is very low. In the dynamic ones, since the logo and body image are found based on the title and text, they are very accurate and realistic. However, due to discussed problems, it was most of the time not possible to get a decent output with this approach. The only feasible solution to this problem was switching to static templates. As a result of the shift from dynamic to static templates, the dynamically generated emails were not subject for testing.

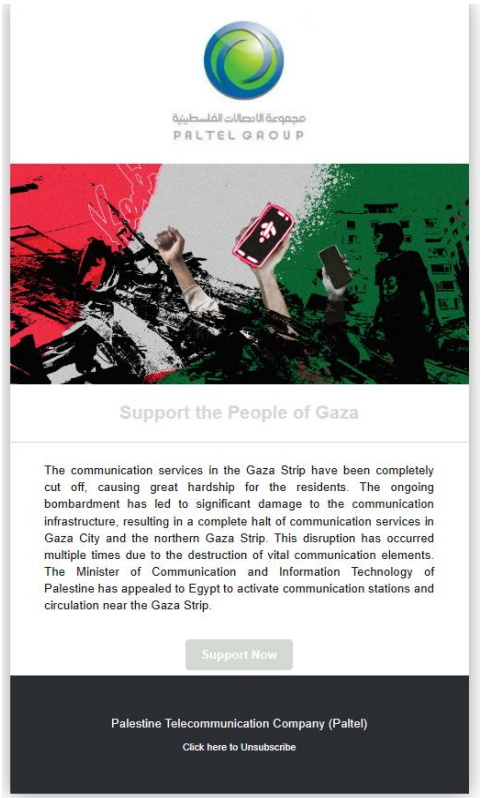

Figure 20. Dynamic template

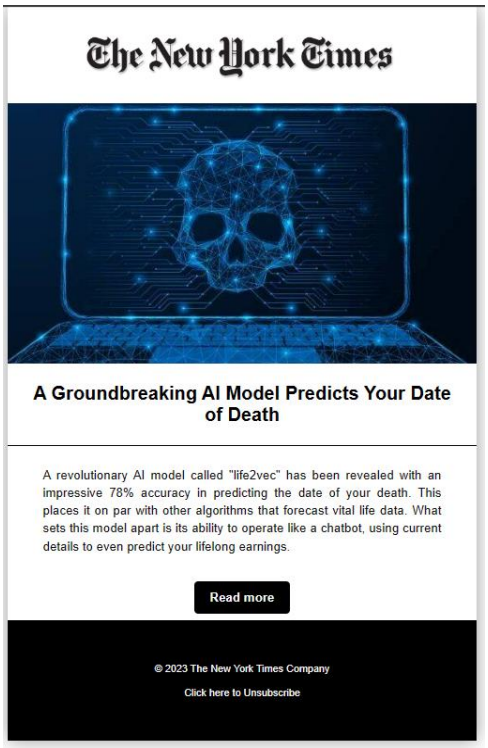

Figure 21. Static template

The testing phase of the news-based phish generator tool was conducted on hundred generated emails. Due to the nature of this tool, the previous testing procedure could not be applied as there was no base sample to compare with. Therefore, all 100 emails were visually inspected and labeled as convincing or unconvincing. As a result, 62 emails among 100 generations were labeled as ‘sufficiently convincing’ by the development team.

However, it was still not enough to conclude the project based on the observations of the team members. Therefore, it is planned to conduct an experiment involving human participants in near future. This experiment will be aimed to evaluate participants’ ability to distinguish between authentic and AI-generated emails. Individuals will be presented with a series of emails without prior knowledge of whether they are AI-generated or not. Participants then will be asked to determine whether each email was generated by Smart Phish Generator or not. The results of the experiment will show if the project was successfully developed.

5. IMPACT

Smart Phish Generator has a high potential to be used in cybersecurity applications. As discussed earlier, studies show that advancements in LLMs were reported to play a significant role in generating convincing phishing mails, which results in anti-phishing tools failing to detect such emails. This project may have a positive impact on improvements in anti-phishing tools as it generates phishing emails and discusses the techniques used. Additionally, the results contribute to understanding human vulnerabilities and the effectiveness of phishing attacks, which may be beneficial in future social engineering studies. Moreover, this project demonstrates the benefits of leveraging AI tools in software systems as it showcases the potential of machine learning in automation tasks. Utilizing from an LLM in the middle of a program, especially when it is not easy to be handled by hardcoded methods, will be inevitable in future projects. Since the html content manipulation was mostly done by LLM’s, this project itself can be considered as a decent example of embedding AI-powered tools in software architecture.

Although this project is not going to be commercialized, it is worth mentioning that all of the technologies used in the project are free from legal restrictions. The only legal restriction that was encountered during the project was using company logos while

generating the phishing emails. This issue was solved by either manipulating the logo or creating a new mock logo for the company.

6. ETHICAL ISSUES

As the nature of this project is dangerous in the context of fraudulent and malicious activities, there were some ethical concerns that needed to be addressed. First of all, as mentioned in Security Constraints, it was and will be made sure that this project will not be involved in any malicious activities. For such purpose, it is planned to implement strict security measures on both the codebase and the final product of the project to prevent unauthorized access to the tool. This will hopefully ensure that the project will not exceed the academic domain.

Another ethical consideration is the use of personal data within the project. To address this, it was ensured that personal information in collected emails was not used, and kept secure accordingly during the project.

Moreover, it was guaranteed that the project was conducted in accordance with all relevant laws and regulations related to data privacy, cybersecurity and copyrights. It is also worth noting that informed consent from participants involved in the study were obtained, and clear information about the nature of the research and how their data would be used was provided.

7. PROJECT MANAGEMENT

At the beginning of the project, the primary objective was to develop an AI-based phishing generator that converts the given authentic email to a phishing email by modifying the components without fully changing them. Considering the final product, it is evident that the intended objective was achieved. However, there were minor changes occurring during the implementation of the project.

In the early phases of the project, the model was programmed to create an html file having a similar template with the input file, then the aim was to fill the template fields with content to create a new context. Aiming to create a template similar to input html, a simple responsive template generator framework was developed and used. However, there were

problems regarding the persuasiveness of generated emails using this tool, as it lacked the characteristic variations that emails could have. In other words, all email templates generated with this tool were looking very similar to each other. Therefore, another technique was developed. Following this technique, all the modifications were started to be done in-place, i.e., that the content of the components in the input emails were replaced with newly generated content.

Another difference between initial and final plans was the modification of logos and main images. Initially, the logo of the output email was determined to be the logo of the input email with small changes (e.g., letter replacement), whereas the main image of the output email was determined to be selected from Google Images with proper queries. However, in order to have a final product that was fully utilized Artificial Intelligence, another technique utilizing Dall-E to generate images and logos were proposed. As discussed in above sections, a hybrid method to randomly decide the image manipulation technique was concluded to be used since none of the methods was determined to be superior to the other.

Additionally, until the late phases of the project, phish generator was being tested with emails as inputs. However, all those generated emails were either imaginary or written using old data, since the AI model is not aware of events that happened after 2021. Hence, news-based phishing emails aim to be more realistic because they are neither imaginary nor written with old data.

Managing the project was a comprehensive learning experience. Working within time pressure was beneficial for organizing tasks efficiently and manage the time better. Additionally, teamwork skills for each member was improved. Collaborative efforts of team members and weekly meetings with the supervisor greatly contributed to the project's success, while improving the communication skills. There were several conflicts between team members during the process that challenged the management of the project. Nevertheless, team members brainstormed their ideas in such conflicts and collectively selected the best option in favor of the project.

8. CONCLUSION AND FUTURE WORK

Having explained the whole process of the project and discussed the results of the developed model, it can be concluded that an automated program that converts authentic emails to phishing emails was successfully developed, yet further developments are needed in the scope of this project. Firstly, model should be aimed to generate even more persuasive emails. Different prompts and queries may lead to more accurate images, which may increase the persuasiveness of the email. Moreover, it should be aimed to generate texts that seem more human-written. In order to achieve this, more analysis on ChatGPT may be done. Also, the developments in the alternative LLMs such as Meta's Llama or Google's Gemini should be taken into consideration, and comparisons between the models should be made to select the model giving the best outputs.

Secondly, the model should be enriched in terms of the subjects it uses to generate phishing emails. As of now, there are 9 loss-based, 8 reward-based and 5 neutral subjects. New subjects should be added to the model to widen the range of the outputs model generate. On the other hand, in addition to approximately 100 email samples that were used as inputs during the project, new authentic emails should be collected and used in order to detect possible errors and bugs that the model might have.

Finally, the testing phase should be expanded in terms of the spectrum of output emails, and number of participants involved. Whilst increasing the number of outputs will lead to have a better understanding of the subjects that are more persuasive and attractive to readers, increasing the number of participants will strengthen the reliability of the test results. Moreover, analyzing the results by grouping the participants (e.g. by their age, gender etc.) might allow finding deep patterns about the email type that each group is most vulnerable to be tricked.

9. REFERENCES

Karim, A., Azam, S., Shanmugam, B., Kannoorpatti, K., & Alazab, M. (2019). A Comprehensive Survey for Intelligent Spam Email Detection. *IEEE Access, 7*, 168261-168295.

Merwe, A. v. d., Marianne, L., and Marek, D. (2005). "Characteristics and responsibilities involved in a Phishing attack, in WISICT '05: proceedings of the 4th international symposium on information and communication technologies. Trinity College Dublin, 249–254

APWG (2020). APWG phishing attack trends reports. 2020 anti-phishing work. Group, Inc. Available at: https://apwg.org/trendsreports/ (Accessed April 30, 2023).

Gupta BB, Arachchilage Nalin AG, Psannis KE. Defending against phishing attacks: taxonomy of methods, current issues and future directions. Telecommun Syst. 2018;67(2):247-267

Hazell, J. (2023). Large Language Models Can Be Used To Effectively Scale Spear Phishing Campaigns.

Roy, S., Naragam, K., & Nilizadeh, S. (2023). Generating Phishing Attacks using ChatGPT.

J. Williams & Danielle Polage (2019) How persuasive is phishing email? The role of authentic design, influence and current events in email judgements, Behaviour & Information Technology, 38:2, 184-197, DOI: 10.1080/0144929X.2018.1519599

Kim, D., & Hyun Kim, J. (2013). Understanding persuasive elements in phishing e-mails: A categorical content and semantic network analysis. *Online Information Review, 37*(6), 835-850.