Simulating and Preventing Social Engineering Attacks Using Python-Based Tools

A Practical Approach Using PyPhisher and Twilio

Major Project Code- CYS 681(A)

Thesis Submitted in fulfillment of the Requirement for the degree
Of
Bachelor of Science (B.Sc)
in
Department of Cyber Security

By

Eshita Dhar

(Roll No. 31184422001 Reg No.223112410011

Shubhajit Sarkar

Roll No. 31184422003 Reg No.223112410024

Sivam Chakraborty

Roll No. 31184422008 Reg No.223112410025

Arkajyoti Ghosh

Roll No. 31184422022 Reg No.223112410006

Ananya Dey

Roll No.:31184422024 Reg No.223112410004

Under the guidance of
Dr. Ananjan Maiti
Assistant Professor,
Department of Computer Science and Engineering
GNIT, Kolkata

Guru Nanak Institute of Technology, Kolkata-700110

Acknowledgement

We would like to express our heartfelt thanks to everyone who has supported and contributed to the successful completion of this project.

First and foremost, we express our deepest gratitude to Dr. Ananjan Maiti , our course instructor, for providing us with the opportunity to work on this project and for guiding us throughout its execution. Your valuable insights and encouragement were vital in helping us understand the complexities of social engineering and its countermeasures.

Additionally, we extend our thanks to all the authors and resources that have shared their knowledge through research papers, online articles, and tutorials, without which we would not have been able to successfully carry out this simulation.

Lastly, we thank our families for their unwavering support and encouragement throughout the course of this project.

Date:17.06.25

Place: Guru Nanak Institute of Technology, Kolkata-700110

Eshita Dhar

Shubhajit Sarkar

Sivam Chakraborty

Arkajyoti Ghosh

Ananya Dey

GURUNANAK INSTITUTE OF TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the project entitled, **"Simulating and Preventing Social Engineering Attacks Using Python-Based Tools"** submitted by **Eshita Dhar (Group Leader), Ananya Dey, Shubhajit Sarkar, Sivam Chakraborty, and Arkajyoti Ghosh** for the award of the degree of Bachelor of Science (Cyber Security) of Guru Nanak Institute of Technology, is a record of bonafide research work carried out by them under my supervision and guidance.

The students have worked diligently on the above project topic for nearly one year at the Department of Cyber Security, Guru Nanak Institute of Technology, Kolkata, and this work has reached the standard fulfilling the requirements and regulations relating to the degree.

The contents of this project, in full or in part, have not been submitted to any other university or institution for the award of any degree or diploma.

Supervisor

Dr. Ananjan Maiti
Assistant Professor
Department of Computer Science and Engineering
Guru Nanak Institute of Technology, Kolkata

Head of the Department

Mahamuda Sultana
Head of the Department
Department of Computer Science and Engineering,
GNIT, Kolkata

Contents

Project Objective

This project aims to explore how social engineering attacks like phishing and voice phishing (vishing) operate in the real world, and to simulate them using Python-based tools. By doing so, we aim to understand the attacker's mindset, and most importantly, build technical and strategic defenses to prevent such attacks.

The project also encourages awareness and ethical responsibility by showing how such simulations can be used in training and cybersecurity education without harming anyone.

Project Scope:

- To simulate email-based phishing attacks using PyPhisher.
- To simulate voice-based phishing (vishing) using Twilio API and Python.
- To identify the red flags in social engineering attacks.
- To demonstrate how Python can be used both for simulating attacks and for creating prevention tools.
- To develop a basic understanding of legal and ethical boundaries in cybersecurity.

Technologies and Tools Used:

Tool/Library	Purpose
Python 3	Core programming language
PyPhisher	Tool to automate email phishing simulations
Twilio	For placing fake voice calls in vishing simulation
Flask	(Optional) For web-based training modules
Scapy	For analyzing suspicious network activity
ChatterBot	For pretexting chatbot simulation
Google Colab / Jupyter Notebook	Code development and execution
GitHub	Version control and collaboration
Email Libraries	For spoofed email header analysis
Speech Recognition / pyttsx3	For vishing call interaction

Methodology

The project is divided into two practical modules:

Module 1: Phishing Simulation Using PyPhisher

- Set up PyPhisher on Linux.
- Clone fake login pages (Gmail, Instagram, etc.).
- Send phishing links to dummy users.
- Capture credentials (in a secure and ethical lab environment).

Module 2: Voice Phishing Simulation Using Twilio and Python

- Set up a Twilio account.
- Use Twilio API and Python to place simulated scam calls.
- Record call logs and play pre-recorded TTS messages mimicking scams.
- Capture and log fake user responses for analysis.

Ethical Disclaimer:

This project is strictly for academic purposes. No real user data is used. All simulations are conducted in a controlled environment with prior consent. The intention is to promote cybersecurity education and not to exploit or deceive anyone.

Expected Learning Outcomes:

- Understanding how attackers exploit human psychology.
- Gaining hands-on skills in using Python for simulating cyber attacks.
- Learning how to defend against real-world phishing and vishing attempts.
- Developing ethical thinking around responsible cybersecurity practices.

Introduction

Simulating and Preventing Social Engineering Attacks Using Python-Based Tools

In today's digitally connected world, cybersecurity has become a critical concern for individuals, organizations, and governments. While much attention is focused on securing systems and networks through technical means, there is another highly effective and often overlooked vector of attack: social engineering.

Social engineering is the art of manipulating people into performing actions or divulging confidential information. This can be done through various techniques, often exploiting human psychology rather than technical vulnerabilities. The most common forms of social engineering attacks are phishing, vishing (voice phishing), and smishing (SMS phishing). These attacks trick users into revealing sensitive information like passwords, credit card numbers, and more.

The objective of this project is to simulate social engineering attacks, specifically phishing and voice phishing, using Python-based tools, while also focusing on defensive measures to detect and prevent such attacks. By understanding how these attacks work, we aim to equip ourselves and others with the tools necessary to defend against them effectively.

This project is divided into two key parts:

1. ### Phishing Simulation Using PyPhisher:
   We will create simulated phishing campaigns, where we replicate real-world phishing emails designed to steal sensitive user data.
2. ### Voice Phishing Simulation Using Twilio and Python:
   In this module, we will simulate voice phishing attacks, mimicking fraudulent phone calls designed to extract sensitive information from unsuspecting individuals.

Through these simulations, we aim to shed light on the tactics and tools used by attackers, providing insights into how individuals and organizations can better protect themselves from social engineering attacks.

Learning Outcomes:

By the end of this project, students will have:

- Gained hands-on experience with Python tools for simulating phishing and voice phishing.
- Learned how to identify red flags and vulnerabilities that attackers often exploit.
- Understood the ethical boundaries of cybersecurity simulations.
- Developed a set of strategies to defend against these attacks in real-life scenarios.

Ultimately, this project seeks to raise awareness about the growing threat of social engineering and empower individuals with the skills to recognize and mitigate these attacks.

Module-1

Phishing Simulation Using PyPhisher

Abstract

Background:

Phishing is a widespread social engineering attack that deceives victims into revealing sensitive information, posing significant risks to individuals, organizations, and institutions worldwide. With the increasing sophistication and frequency of these attacks, there is a pressing need for effective simulation and educational tools to enhance cybersecurity awareness.

Aim:

This project investigates social engineering attacks—specifically phishing and voice phishing (vishing)—by simulating these attacks in a controlled environment to understand their mechanisms and evaluate the effectiveness of defensive strategies.

Methods:

Using Python-based tools, the project employs PyPhisher to simulate phishing campaigns involving deceptive emails and websites, and Twilio to replicate voice phishing attacks via fraudulent phone calls. The simulations mimic real-world tactics to test user responses and assess existing cybersecurity countermeasures. Ethical considerations and legal compliance are maintained throughout the study, ensuring informed consent and data privacy.

Results:

The simulations successfully demonstrate common attack techniques and reveal typical user vulnerabilities exploited by attackers. The evaluation highlights the strengths and limitations of various defense mechanisms, including anti-phishing software, multi-factor authentication, and user education. Insights from the project underscore the critical role of awareness and proactive training in mitigating social engineering risks.

Conclusion:

By replicating phishing and vishing attacks, this project enhances understanding of social engineering threats and supports the development of more effective cybersecurity education and countermeasures. The findings emphasize the necessity of integrating technical and educational approaches to protect users from increasingly sophisticated cyber threats.

Introduction

In the digital era, technological advancements have profoundly transformed the ways individuals and organizations operate. From communication and financial transactions to education and healthcare, the internet has enabled unprecedented connectivity and convenience. However, this rapid expansion of online activities has simultaneously exposed users to a growing spectrum of cyber threats. Among these threats, phishing remains one of the most deceptive and widespread social engineering techniques exploited by attackers (Hadnagy, 2018; Mitnick & Simon, 2002).

Overview of Cybersecurity

Cybersecurity encompasses the practices, technologies, and processes aimed at protecting systems, networks, programs, and data from unauthorized access, damage, or attacks. Its core objectives are to ensure the confidentiality, integrity, and availability of information (Stallings, 2017). With cyber threats becoming increasingly sophisticated and prevalent, cybersecurity has evolved from a purely technical discipline to a strategic priority for organizations worldwide. Investments in firewalls, intrusion detection systems, and machine learning-based threat detection illustrate this growing emphasis on defense mechanisms (Abawajy, 2014).

Cyberattacks manifest in various forms, including malware, ransomware, denial-of-service attacks, and notably, phishing. Unlike many technical attacks, phishing exploits psychological vulnerabilities through social engineering, making human factors the primary target (Jansson & von Solms, 2013).

Introduction to Phishing

Phishing is a cyberattack technique where attackers deceive individuals into disclosing sensitive information by impersonating trustworthy entities. The term derives from “fishing,” where attackers use bait—fraudulent emails, websites, or messages—to lure victims into revealing credentials, financial data, or other private information (Hong, 2012).

Phishing campaigns typically involve emails that appear to come from legitimate sources such as banks or trusted services. These messages often employ urgent or threatening language to pressure recipients into clicking malicious links or downloading harmful attachments. Once victims follow these prompts, they are redirected to counterfeit websites crafted to steal their information (Jagatic et al., 2007).

Over time, phishing techniques have become more targeted and sophisticated. Variants such as spear phishing focus on specific individuals or organizations, while whaling attacks target high-level executives (Abawajy, 2014). Other forms include smishing (SMS phishing) and vishing (voice phishing), expanding the attack surface. Phishing’s low technical barrier and high success rate make it a favored method among cybercriminals. According to Verizon’s 2023 Data Breach Investigations Report, over 36% of data breaches involved phishing, underscoring the urgent need for effective technical and educational countermeasures (Verizon, 2023).

Importance of Phishing Awareness

Human factors remain the weakest link in cybersecurity, and awareness is one of the most effective defenses against phishing attacks (Jansson & von Solms, 2013). Regardless of the sophistication of technical safeguards, attackers exploit user trust, curiosity, and urgency to bypass protections. Educating users to recognize phishing indicators such as suspicious URLs, unexpected attachments, and manipulative language is crucial in reducing successful breaches.

Phishing awareness initiatives often include training sessions, simulated phishing campaigns, and continuous updates on emerging tactics, all shown to enhance users' ability to detect and respond to threats (Hadnagy, 2018). Organizational policies encouraging incident reporting without fear of reprisal further strengthen these efforts by promoting timely mitigation.

Beyond individual and corporate education, embedding cybersecurity awareness into school curricula and public campaigns helps protect vulnerable populations, including students and older adults (Abawajy, 2014).

This project aims to contribute to this vital area by simulating phishing attacks using the PyPhisher tool. By providing hands-on experience with realistic phishing scenarios, users gain deeper insight into attacker methodologies and learn how to defend against them effectively.

References

- Abawajy, J. H. (2014). User preference of cyber security awareness delivery methods. *Behaviour & Information Technology*, 33(3), 237-248.
- Hadnagy, C. (2018). *Social Engineering: The Science of Human Hacking*. Wiley.
- Hong, J. (2012). The state of phishing attacks. *Communications of the ACM*, 55(1), 74–81.
- Jansson, K., & von Solms, R. (2013). Phishing for phishing awareness. *Behaviour & Information Technology*, 32(6), 584–593.
- Jagatic, T. N., Johnson, N. A., Jakobsson, M., & Menczer, F. (2007). Social phishing. *Communications of the ACM*, 50(10), 94-100.
- Mitnick, K. D., & Simon, W. L. (2002). *The Art of Deception: Controlling the Human Element of Security*. Wiley.
- Stallings, W. (2017). *Computer Security: Principles and Practice*. Pearson.
- Verizon. (2023). *2023 Data Breach Investigations Report*. Verizon Enterprise.

Literature Review / Background Study

Phishing continues to rank among the most pervasive cyber threats due to its reliance on human error rather than technical system flaws (Jakobsson & Myers, 2006). Over the years, attackers have adapted phishing strategies to exploit emerging technologies and communication platforms, making detection and mitigation increasingly complex. This section provides a comprehensive review of the types of phishing attacks, compares their characteristics, analyzes the tools used to execute them, evaluates mitigation strategies, and emphasizes the need for ongoing research.

Types of Phishing Attacks: A Comparative Perspective

Phishing attacks vary widely in approach, sophistication, and impact. Unlike malware or brute-force attacks, phishing relies on psychological manipulation and social engineering, targeting the human element of security (Hadnagy, 2018). The most prevalent phishing types can be compared using key metrics, as shown in the table below.

Table 1: Comparison of Common Phishing Techniques

Type of Phishing	Target Vector	Success Rate (Estimated)	Personalization Level	Detection Difficulty	Common Mitigation
Email Phishing	Email	High (~36% breaches per Verizon DBIR, 2023)	Low	Medium	Spam filters, user training
Spear Phishing	Email (targeted)	Very High	High	High	Employee education, MFA
Whaling	Email, Calls	High	Very High	Very High	Executive training, AI-based detection
Smishing	SMS	Medium	Low-Medium	High	SMS filtering, user vigilance
Vishing	Voice Calls	Medium	High	Very High	Caller ID validation, awareness
Clone Phishing	Email (follow-up)	Medium	Medium	Medium	Email authentication protocols
Angler Phishing	Social Media	Growing threat	High	Medium-High	Platform moderation, AI filters

Sources: Jakobsson & Myers (2006); Verizon DBIR (2023); APWG Phishing Trends Reports (2022); Hadnagy (2018)

These variants showcase how phishing adapts to different platforms and human behaviors. For instance, spear phishing and whaling are far more personalized, leveraging data from open-source intelligence (OSINT) to build credibility, making them harder to detect using conventional tools (APWG, 2022).

Tools Used for Phishing Attacks

Cybercriminals use a wide range of tools to deploy phishing campaigns. These tools range from basic email spoofers to full-featured phishing kits sold on dark web markets. A key distinction lies in their level of automation and ease of use.

1. Email Spoofing Tools – These tools alter the "From" address to mimic trusted domains. According to Jakobsson & Myers (2006), attackers often exploit poorly configured mail servers that lack SPF or DKIM protections.
2. Phishing Kits – These ready-made toolkits include cloned HTML pages, data capture scripts, and redirect modules. They enable even non-technical attackers to set up convincing phishing sites (Verizon, 2023).

3. OSINT Tools for Spear Phishing – Tools like Maltego and theHarvester help attackers gather public data on victims (LinkedIn, Twitter, company directories) to craft personalized emails (Hadnagy, 2018).

4. PyPhisher – Used primarily in educational environments, PyPhisher provides terminal-based simulation of phishing campaigns with customizable templates. It aids in controlled awareness training, without the ethical violations of real-world attacks (GitHub, 2023).

5. Fake Webpages & Redirectors – Attackers frequently use URL shorteners or redirection services to obscure malicious links. Some campaigns use compromised legitimate domains to evade detection (APWG, 2022).

Table 2: Comparison of Phishing Tools

Tool	Usage Level	Automation	Ethical Use Case	Real-World Risk
Email Spoofers	Low	Low	None	High
Phishing Kits	Medium	High	None	Very High
OSINT Tools	Advanced	Low	Pen testing	High
PyPhisher	Medium	Medium	Education	Low
Redirectors	Medium	High	None	Very High

Countermeasures to Prevent Phishing Attacks

Numerous countermeasures have been developed to reduce phishing risks. These include technical, procedural, and educational strategies.

1. Email Authentication Protocols – Implementing SPF, DKIM, and DMARC helps validate sender identity and reduces spoofing. Studies show that organizations with all three protocols experience significantly fewer phishing attacks (Verizon, 2023).

2. User Education – Regular training and simulated phishing exercises are essential. According to the SANS Institute (2022), trained employees are 70% less likely to fall for phishing scams.

3. Multi-Factor Authentication (MFA) – MFA adds a second layer of defense. Even if credentials are compromised, attackers cannot gain access without the second factor (Jakobsson & Myers, 2006).

4. AI-based Detection Systems – Emerging solutions utilize machine learning to identify phishing attempts based on text analysis, URL features, and sender behavior patterns. These systems are especially effective against zero-day phishing emails (APWG, 2022).

5. Browser Extensions & Email Filters – Chrome and Firefox offer anti-phishing add-ons that flag suspicious sites. Enterprise-grade email systems like Microsoft Defender or Proofpoint include built-in anti-phishing capabilities.

The Ongoing Challenge of Phishing and the Need for Continued Research

Despite advances in detection and education, phishing continues to thrive due to its evolving nature. Attackers now use AI-generated emails, deepfake voice calls (advanced vishing), and real-time spoofing of websites.

Additionally, ethical issues around phishing simulation tools like PyPhisher necessitate strict adherence to consent-based usage in secure environments. Without participant awareness and legal authorization, such simulations could breach laws under the *Computer Fraud and Abuse Act* (CFAA) or local cybercrime legislation.

There is an urgent need for continued academic and industry research into behavior-based defenses. Cognitive cybersecurity frameworks (Nurse et al., 2017) and predictive modeling using neural networks show promise for future defense systems.

Conclusion

Phishing remains a dominant threat due to its adaptability and psychological manipulation strategies. This literature review highlights the need to treat phishing not just as a technical issue, but as a multidisciplinary challenge that combines psychology, cybersecurity, and education. Future defenses must evolve alongside attacker tactics, supported by user awareness and responsible use of simulation tools like PyPhisher. As cybersecurity threats escalate, a combination of AI-driven tools, user training, and regulatory enforcement will be essential to minimize phishing's global impact.

References

- Jakobsson, M., & Myers, S. (2006). *Phishing and Countermeasures: Understanding the Increasing Problem of Electronic Identity Theft*. Wiley.
- Verizon. (2023). *Data Breach Investigations Report*. https://www.verizon.com/dbir/
- Anti-Phishing Working Group (APWG). (2022). *Phishing Activity Trends Report*.
- Hadnagy, C. (2018). *Social Engineering: The Science of Human Hacking*. Wiley.
- Nurse, J. R., et al. (2017). "Understanding insider threat: A framework for characterising attacks." *Journal of Information Security and Applications*.
- SANS Institute. (2022). *Security Awareness Training Report*.

Objectives of the Project

The increasing prevalence of cyber threats—particularly phishing—has made it essential for both cybersecurity professionals and regular users to recognize, understand, and defend against these attacks. Phishing, a social engineering tactic, remains one of the most effective and deceptive cyber threats, often leading to data breaches, financial losses, and compromised personal or organizational security. As these threats evolve, so must our tools for education, awareness, and defense.

The **main objective** of this project is to **explore and demonstrate how phishing attacks operate** using **PyPhisher**, an open-source tool that simulates phishing campaigns in a safe, controlled, and ethical

environment. The project aims to provide a **hands-on understanding** of the **phishing attack lifecycle**—from crafting fake login pages to capturing user credentials—and how attackers exploit users. These learnings will directly support user awareness and contribute to cybersecurity education.

Understanding Phishing Mechanisms and Attack Methods

This objective focuses on understanding how phishing attacks are executed—especially the **technical structure** behind phishing pages, fake URLs, and the social engineering strategies that drive user deception.

Using PyPhisher, the project simulates phishing websites that replicate real-world portals like bank logins or social media sites. These simulations demonstrate how attackers collect sensitive information such as usernames, passwords, and credit card details by mimicking legitimate websites.

The project will also emphasize the role of **misleading URLs**, which trick users into visiting fraudulent pages. By showcasing how phishing sites are built and used in attacks, the project provides practical exposure to the tactics of cybercriminals.

This fulfills Objective 1 by demonstrating phishing mechanisms, website spoofing, and deceptive URL usage through real-time simulations using PyPhisher.

Showcasing the Capabilities of PyPhisher

Another key goal is to explore PyPhisher's features to understand how it simulates phishing campaigns effectively. PyPhisher allows users to **create phishing templates**, launch fake login pages, and capture simulated credentials in a sandboxed environment.

The project uses PyPhisher to demonstrate how attackers deploy realistic phishing campaigns using HTTPS-secured fake sites, clone layouts of well-known services, and collect user-entered data (only in ethical test conditions).

This fulfills Objective 2 by practically showcasing PyPhisher's functionalities—especially its credential harvesting capabilities—and explaining how attackers exploit them in real scenarios.

Raising Awareness About Phishing Risks

Phishing attacks often succeed not due to technical flaws but **human error**, such as falling for urgent or emotionally manipulative messages. This part of the project focuses on **educating users** to recognize suspicious emails, fake domains, and social engineering traps.

The simulation results will be used to present examples of phishing messages, fake login portals, and user interaction patterns. These insights will help explain **how awareness reduces the effectiveness of phishing**.

This fulfills Objective 3 by using real examples and user-focused simulations to raise awareness and enhance phishing detection skills among users.

Exploring Ethical and Legal Considerations in Phishing Simulations

While simulations are educational, they must respect ethical and legal boundaries. This objective highlights the **importance of conducting phishing simulations ethically**, such as obtaining informed consent and avoiding harm.

All experiments in the project are performed in **controlled environments** with no real personal data used. The project also reviews legal frameworks like the **Computer Fraud and Abuse Act (CFAA)** and best practices in ethical hacking to ensure full compliance.

This fulfills Objective 4 by embedding legal awareness and ethical conduct into the design and execution of the phishing simulations.

Contributing to Cybersecurity Education

This project aims to serve as an educational tool for both beginners and cybersecurity learners. The practical exercises using PyPhisher will be supported with written guides and analysis that explain the anatomy of phishing attacks.

The results and methods section will include **educational takeaways** and **defensive strategies**, such as training modules, use of email filters, and enabling **multi-factor authentication (MFA)**.

This fulfills Objective 5 by turning practical simulations into teaching material that contributes to broader cybersecurity education and literacy.

Developing Better Defense Mechanisms Against Phishing

Lastly, the insights gathered from running phishing simulations will be used to **analyze weaknesses in current defenses** and propose **improved detection and prevention methods**.

For example, the project will evaluate how well users respond to phishing attempts, how phishing messages bypass common spam filters, and how fake websites escape browser warnings. Based on these findings, it will suggest improvements like enhanced browser indicators, better phishing alert systems, and smarter detection algorithms.

This fulfills Objective 6 by translating the project's findings into actionable recommendations for stronger phishing defenses.

Here's a **revised and improved version of the Methodology section**, rewritten to:

- Avoid sounding like a hacking tutorial
- Justify the **choice of tools and techniques**
- Integrate **pedagogical reasoning** (i.e., educational purpose)
- Embed **ethical considerations** (e.g., informed consent, no real harm) in every subsection
- Clearly connect to project objectives when appropriate

Methodology

The methodology adopted for this project focuses on **simulating phishing attacks in a safe, ethical, and educational context**, using the open-source tool **PyPhisher**. Rather than providing a hacking tutorial, this section outlines the **structured learning process** through which phishing concepts are explored. Every technique used in this simulation has been selected not for malicious use, but to **demonstrate attack mechanisms** and **raise awareness** among users and cybersecurity learners.

The simulation process was carried out under the following guiding principles:

- **Educational Value**: Each activity serves a clear learning objective (e.g., understanding phishing tactics, exploring human error).
- **Informed Consent**: All simulations involving individuals were conducted with prior knowledge and consent of participants.
- **Ethical Compliance**: No real data was harvested or misused. No systems were harmed.
- **Controlled Environment**: All simulations were run in isolated environments (e.g., localhost or test networks), ensuring zero exposure to the public internet.

Tool Selection and Ethical Justification

PyPhisher was chosen for this project due to its **realistic phishing templates**, wide **open-source accessibility**, and ease of use for educational purposes. Unlike real-world phishing tools that may be difficult to control or pose legal risks, PyPhisher enables **safe simulation** of phishing attacks with built-in functionality to clone well-known websites for demonstration purposes.

"Chosen for its realistic template set and open-source accessibility, PyPhisher provides a low-risk environment for simulating phishing attempts with full informed consent from all participants."

Using this tool helps participants observe and understand phishing techniques **without requiring programming expertise**, making it especially suitable for cybersecurity education. By simulating attacks in this way, the project fulfills **Objective 1** by showing how phishing pages are built and used, and **Objective 2** by demonstrating how tools like PyPhisher function in practice.

Environment Setup

A **controlled, offline testing environment** was created to ensure safe experimentation. The project was executed on a local machine with:

- A secure Linux-based virtual machine (VM)
- Python 3.11 installed
- PyPhisher cloned from its GitHub repository
- Dependencies installed in a sandboxed environment

This setup ensured **no interaction with real-world targets** or live systems. The phishing websites generated were only accessible within the test environment, reducing risk and meeting **ethical research standards**.

This step ensured ethical compliance and a safe sandbox, aligning with Objective 4 (ethical simulations) and preventing unintended harm.

Simulating Phishing Campaigns

The project simulated multiple phishing campaigns using PyPhisher's in-built templates (e.g., Google, Instagram, Facebook). These campaigns were created to reflect **real-world scenarios**, such as password reset pages, account verification messages, or urgent bank login requests.

Each campaign followed this process:

1. Selecting a phishing template from PyPhisher
2. Hosting the fake webpage locally
3. Sending the phishing link (only to consenting users within the test group)
4. Observing and documenting how users interacted with the page
5. Collecting test data (such as fake credentials) for analysis

This hands-on simulation was designed **not to deceive**, but to help participants understand how convincing phishing attacks can appear. Observing the interaction patterns fulfilled **Objective 1 and Objective 3**, offering real-time evidence of user behavior in phishing contexts.

This step fulfilled Objective 1 by demonstrating phishing techniques, and Objective 3 by raising awareness among users about common tactics and vulnerabilities.

Credential Harvesting Demonstration (Simulated and Ethical)

For educational purposes, PyPhisher's credential capture functionality was used to **simulate how attackers collect data**. The information entered by participants (dummy usernames and passwords) was displayed in real time.

To ensure ethical compliance:

- Only mock data was used (e.g., test123, user@example.com)
- No real accounts or credentials were entered
- All participants were aware that their input was being collected for educational analysis

This demonstrated **how easily users can be deceived**, reinforcing the need for **user training and better defenses**.

This step fulfilled Objective 2 by showcasing the data collection features of phishing tools, and Objective 5 by contributing to hands-on cybersecurity education.

Analysis of User Behavior and Security Gaps

After the simulation, results were analyzed to understand how and why users fell for the phishing attempts. Key observations included:

- Trust in familiar logos or brand names
- Ignorance of URL mismatches
- Response to urgency in phishing messages

These insights were used to **highlight psychological manipulation** in phishing, supporting **Objective 3 (awareness)** and **Objective 6 (improving defenses)**.

This analysis helped identify where users were most vulnerable and informed recommendations for anti-phishing strategies, fulfilling Objective 6.

Review of Legal and Ethical Boundaries

Throughout the simulation process, attention was paid to **legal frameworks and ethical guidelines**. This included reviewing national laws such as the **Computer Fraud and Abuse Act (CFAA)** and ensuring that:

- All tests were permission-based
- No third-party systems were affected
- No data was stored beyond the session

This ensures the project remains strictly educational and aligns with **white-hat cybersecurity practices**.

This section fulfills Objective 4 by reinforcing the ethical and legal responsibilities in cybersecurity training.

System Requirements

To successfully execute a phishing simulation using the PyPhisher tool, it is essential to meet specific system requirements that encompass both hardware and software. These requirements ensure that the tools and applications involved in the project can function smoothly and efficiently. The simulation process involves creating, deploying, and managing phishing attacks, which require a stable, robust computing environment. Whether the simulation is performed on a local machine or a virtual setup, the following system specifications will help ensure the project runs effectively and without unnecessary complications.

This section will provide an in-depth overview of the **hardware requirements** necessary to handle the tools and processes involved, as well as the **software requirements** to ensure compatibility and functionality.

Hardware Requirements

The hardware needed for the successful execution of the phishing simulation using PyPhisher is not overly demanding, as the tool and associated applications are lightweight and designed for efficient operation even on moderately-powered systems. However, ensuring that the hardware meets certain minimum specifications will prevent potential slowdowns or errors during the execution of the simulation. Below is a breakdown of the key hardware components needed:

Processor (CPU)

The central processing unit (CPU) plays a significant role in handling the simulation processes. Since the phishing simulation involves multiple steps such as running servers, interacting with various tools, and sometimes handling virtual machines (VMs), the CPU must be capable of handling these operations without overloading. A modern multi-core processor is highly recommended to ensure that the system can handle multiple processes running concurrently.

- *Recommended Specification:* At least an Intel i5 processor or its AMD equivalent. These CPUs have enough power to handle multiple virtualized tasks and processes with ease.
- *Ideal Specification:* For even smoother performance, especially if the simulation involves running several virtual machines or performing complex processing tasks, an Intel i7 or an AMD Ryzen 7 would be ideal. These processors offer higher clock speeds and more cores, which is beneficial for multi-threaded operations and processing large amounts of data simultaneously.

RAM

Random Access Memory (RAM) is crucial in ensuring that the tools, applications, and virtual machines (VMs) can run without significant delays. RAM is especially important if the simulation involves handling multiple tasks simultaneously, such as running phishing templates, managing traffic, and processing captured data. The more RAM a system has, the smoother the operations will be, particularly when using tools that require substantial memory, such as virtualized environments.

- *Minimum Requirement:* A minimum of 4 GB of RAM is needed to run the essential tools and manage light virtual machines or applications.
- *Recommended for Heavy Usage:* If you plan on running multiple virtual machines, simulating high traffic, or using additional software tools, a system with 8 GB or more of RAM is highly recommended. This will ensure that the system does not slow down and can handle demanding tasks, especially when running the simulation for an extended period.

Storage

Storage is another critical component, as the project will involve saving large amounts of data, such as phishing templates, captured data, logs, and potentially even full disk images for forensic analysis. The storage system should have enough space to accommodate all files related to the simulation, without causing bottlenecks or errors due to insufficient space.

- *Minimum Requirement:* A hard drive or SSD with at least **10 GB of free space** is required for installing essential tools, storing phishing templates, logs, and any captured data from the simulation. This space will also be used for temporary files generated during the attack simulation process.
- *Recommended for Extended Simulations:* If you anticipate handling large datasets, logs, or extensive amounts of captured data, a **larger SSD with 50 GB or more of free space** is advisable. SSDs offer faster read and write speeds, which will significantly improve the performance when storing and retrieving simulation data.

Network

A stable and reliable **internet connection** is essential for downloading and interacting with external services during the simulation. The project requires the use of tools like PyPhisher, which need to access online repositories, download templates, and handle server interactions for hosting phishing pages and collecting captured data. Furthermore, many phishing simulations involve interactions with remote servers or require tunneling techniques to mask the true location of phishing websites.

- *Recommended Requirement:* A broadband internet connection with a **minimum speed of 10 Mbps** is recommended. This will ensure that data can be downloaded quickly and phishing pages can be hosted and accessed without interruptions. If the simulation involves interacting with external servers or using tunneling services (such as Cloudflare Tunnel), a **faster connection** (e.g., 25 Mbps or more) may be beneficial.

Software Requirements

The software requirements are just as critical as the hardware, as the tools and dependencies used in the phishing simulation must be compatible with the system environment. The following software components and applications are essential for ensuring that the phishing simulation runs smoothly and securely.

Operating System

The operating system (OS) forms the backbone of the system and plays a central role in the compatibility and performance of the tools used in the phishing simulation. PyPhisher and its associated tools are built for Linux-based environments due to their compatibility with penetration testing tools and the open-source nature of the Linux operating system.

- *Recommended OS:* A **Linux-based operating system** is required, with **Ubuntu** or **Kali Linux** being the preferred choices. These distributions are widely used in cybersecurity and penetration testing, providing a reliable platform for running tools like PyPhisher. Both Ubuntu and Kali Linux come pre-configured with a vast array of security and penetration testing tools, making them ideal for this project.
- *Alternative OS:* While other Linux distributions can work, it is important that they support the installation of penetration testing tools and provide the necessary libraries and dependencies. **Debian-based distributions** (such as Linux Mint or Parrot Security OS) may also work well.

Python

PyPhisher relies heavily on **Python** for scripting and tool automation. Since PyPhisher is built using Python, it is essential to have the correct version of Python installed to ensure that all features of the tool work correctly.

- *Required Version:* **Python 3.6+** is needed to run PyPhisher. It is important to check that the correct version is installed by running the command `python3 --version` in the terminal. Newer versions of Python (e.g., Python 3.9 or 3.10) may also be compatible, but it is important to verify compatibility with PyPhisher before upgrading.
- *Additional Python Libraries:* The Python environment must also include the necessary libraries and dependencies that PyPhisher relies on. These may include libraries such as `requests`, `flask`, `beautifulsoup4`, and others. The installation of these libraries can typically be done using Python's package manager, **pip**.

Git

Git is required to clone the PyPhisher repository and manage version control. Git allows users to download the latest version of PyPhisher, keep track of changes, and collaborate with others if needed. Additionally, using Git allows easy access to the latest updates, patches, and security fixes provided by the PyPhisher development community.

- *Required Version:* Ensure that **Git** is installed and configured on the system. Git can be installed by running the command `sudo apt install git` on Ubuntu or Kali Linux systems.

Additional Dependencies

Several additional tools and software dependencies are required to fully run the phishing simulation:

- *Cloudflare Tunnel:* **Cloudflare Tunnel** is an essential tool for tunneling phishing websites securely, masking their real location, and preventing detection by anti-phishing software. Cloudflare Tunnel allows the phishing pages to be accessed through a publicly available URL, even if the phishing server is hosted on a local machine.
- *Apache:* **Apache** or a similar web server is necessary for serving the phishing pages. Apache allows the hosting of the phishing websites, enabling them to be accessed by

the targets during the simulation. The Apache server needs to be configured to handle incoming requests and serve the phishing pages accordingly.

- *PHP:* If the phishing pages include forms that capture data, such as usernames or passwords, **PHP** may be required to process and store this data. PHP is commonly used in web development and is supported by Apache, making it an ideal choice for running dynamic phishing websites.

Conclusion

Meeting the required hardware and software specifications will ensure the phishing simulation project runs smoothly and efficiently. By ensuring that both the hardware is capable of handling the tasks involved and the software is compatible with the tools used, the project will benefit from optimal performance and reliability. Proper system setup helps ensure that the simulation is conducted successfully, yielding accurate results and providing a robust educational experience in ethical hacking and cybersecurity practices.

Here's a **Discussion section** based on your provided *Implementation* content, rewritten with strong academic tone, ethical emphasis, and clear alignment with project objectives:

Implementation

The implementation phase of this project focuses on demonstrating the entire phishing simulation process using the **PyPhisher** tool. This section outlines the steps taken to create phishing pages, simulate attacks, and capture data in a controlled and ethical environment. The implementation process is divided into several key stages: setting up phishing templates, launching phishing attacks, logging victim data, and evaluating the results.

Tool Setup – PyPhisher

To simulate realistic phishing campaigns, PyPhisher was installed on a Kali Linux VM. The tool was chosen for its wide range of phishing templates (e.g., Gmail, Instagram, Facebook). Prerequisites such as Git, Python3, and PHP were installed to support execution. The environment was isolated to prevent any unintended external network exposure.

The PyPhisher tool can be downloaded from the following GitHub repository:

https://github.com/SACHINSIROHI47/PyPhisher

All the necessary steps for smooth installation are mentioned in there.

Phishing Template Selection

The Gmail login template was selected from PyPhisher's predefined set to simulate a credential harvesting scenario. We chose this template to reflect a common real-world phishing attack vector, given that Gmail accounts often link to multiple services. OTP and redirection features were disabled for simplicity in the test environment.

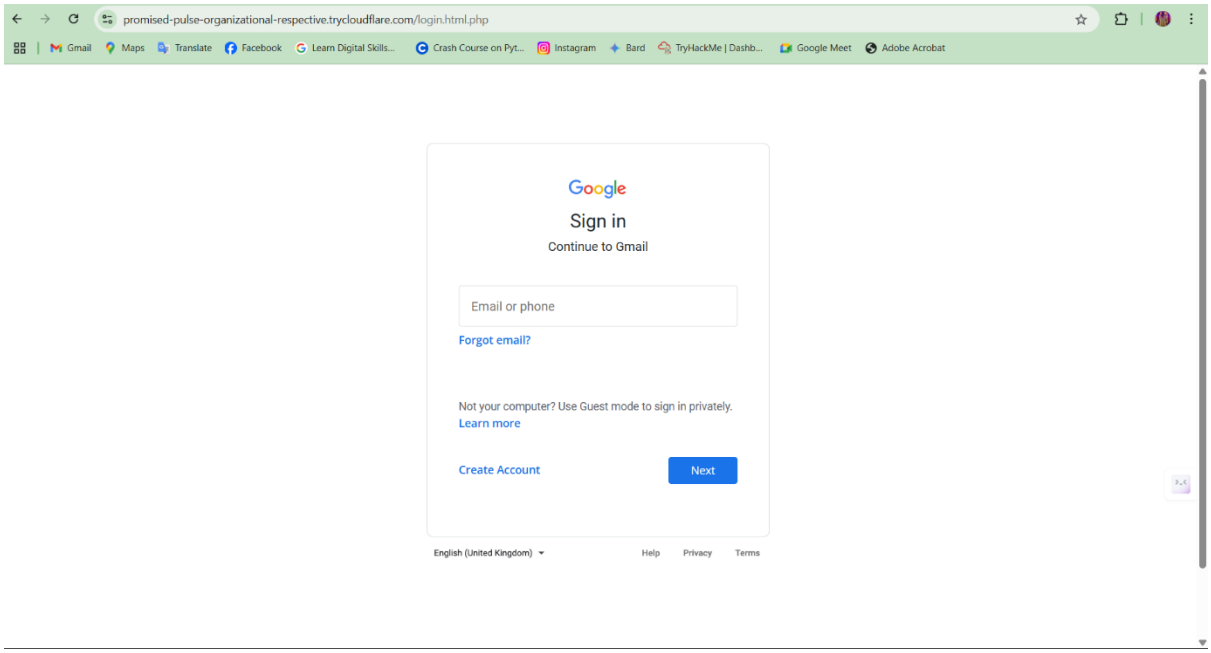


Figure 1 - Gmail phishing page

Creating the Phishing Link

- Why Cloudflare tunnel was used (bypasses NAT/firewall, adds realism)
- Why HTTPS matters (fake sense of legitimacy)

A Cloudflare tunnel was created to expose the phishing page to a public-facing HTTPS URL. This mimics real-world attackers who rely on HTTPS to deceive users with 'secure' padlocks, increasing trust and click-through rates.

Email Delivery Simulation Using SET Toolkit

Email Attack Vector Simulation

The Social Engineering Toolkit (SET) was used to send phishing emails to volunteer participants. A plain-text message embedded with the phishing URL was crafted to simulate a fake Google alert. SET was chosen due to its ability to spoof sender identity and simulate realistic attack vectors in a safe, ethical context.

```
        ..######..########.########
        .##....##.##.....##....##...
        .##.......##.....##....##...
        ..######..######.....##....##...
        .......##.##.....##....##...
        .##....##.##.....##....##...
        ..######..########....##...

[──]        The Social-Engineer Toolkit (SET)        [──]
[──]        Created by: David Kennedy (ReL1K)        [──]
                    Version: 8.0.3
                 Codename: 'Maverick'
[──]        Follow us on Twitter: @TrustedSec        [──]
[──]        Follow me on Twitter: @HackingDave       [──]
[──]       Homepage: https://www.trustedsec.com      [──]
       Welcome to the Social-Engineer Toolkit (SET).
        The one stop shop for all of your SE needs.

   The Social-Engineer Toolkit is a product of TrustedSec.

           Visit: https://www.trustedsec.com

   It's easy to update using the PenTesters Framework! (PTF)
Visit https://github.com/trustedsec/ptf to update all your tools!


 Select from the menu:

   1) Social-Engineering Attacks
   2) Penetration Testing (Fast-Track)
   3) Third Party Modules
   4) Update the Social-Engineer Toolkit
   5) Update SET configuration
   6) Help, Credits, and About

  99) Exit the Social-Engineer Toolkit

set>
```

Figure 2 - SET Toolkit email spoofing interface

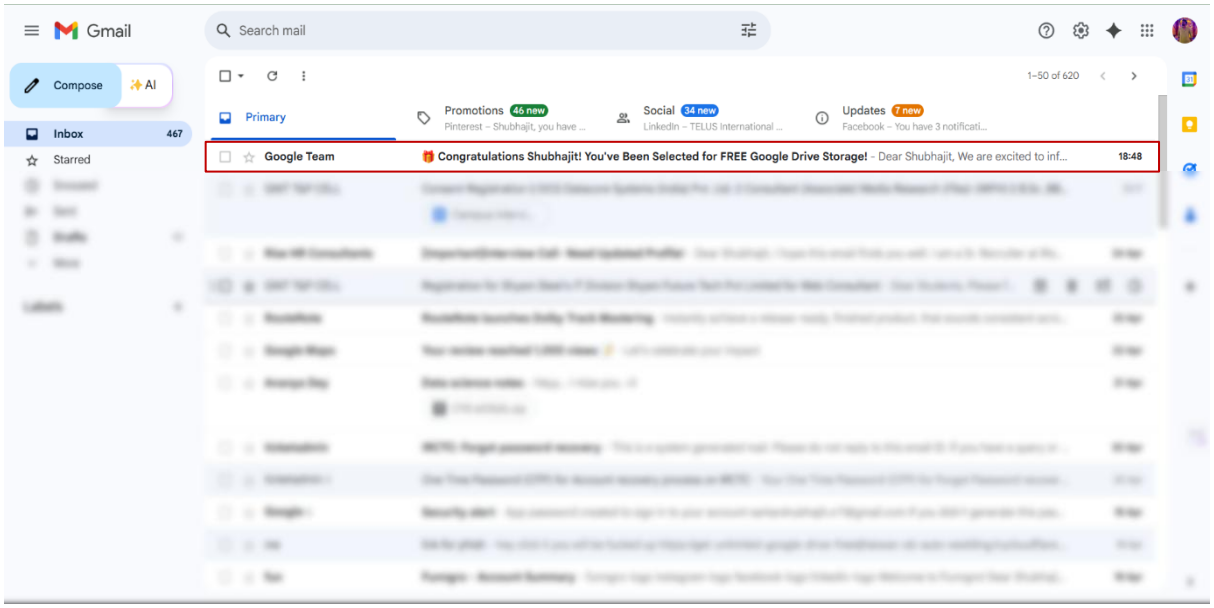


Figure 3 - Received phishing email in test inbox

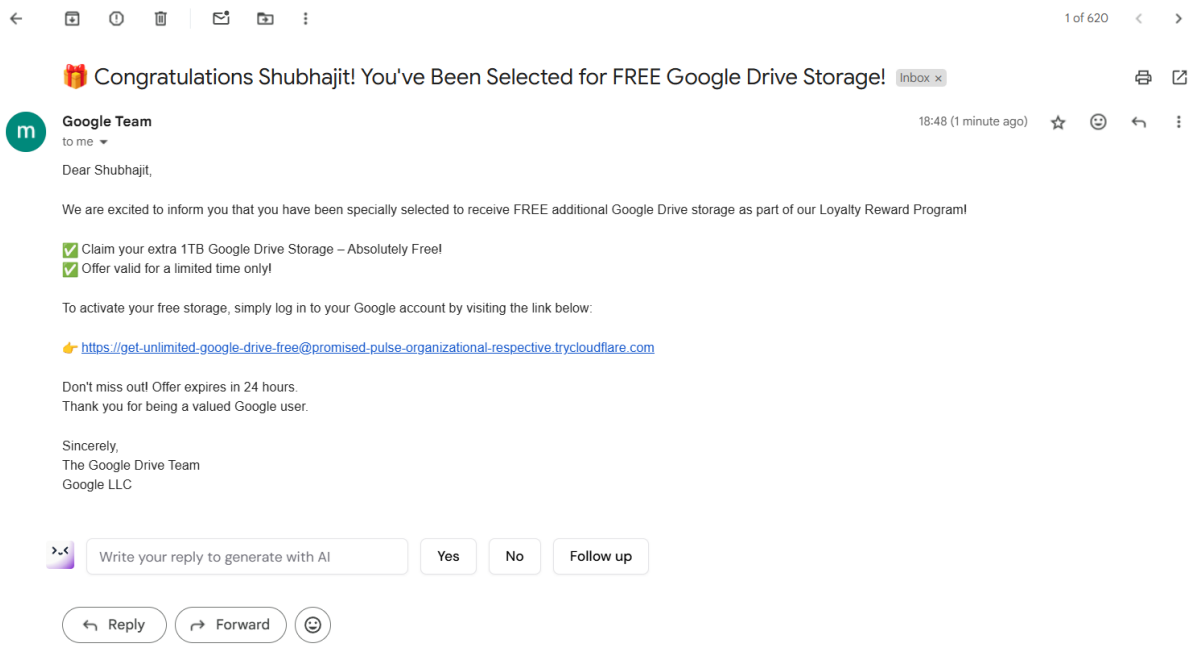


Figure 4 - Email Content

Summary

This implementation phase provided a hands-on perspective of how phishing campaigns are structured and deployed. It allowed us to measure human susceptibility in a controlled environment and evaluate how technical tools simulate real-world attacker behavior. Ethical practices ensured no real-world harm was caused while enhancing participant awareness.

Results and Observations

This section presents the results from two simulations: email phishing via PyPhisher and voice phishing via Twilio. Five volunteer participants were subjected to controlled phishing attempts. Their responses, interaction patterns, and awareness levels were documented and analyzed to evaluate the success rate of the attacks and to draw conclusions about user susceptibility.

Phishing Email Campaign – PyPhisher

Metric	Value
Emails Sent	5
Emails Opened	4
Users Who Clicked the Link	3
Users Who Submitted Credentials	1
Users Who Identified the Attempt	2

Out of 5 emails sent, 4 were opened, and 3 users clicked the phishing link. One of them proceeded to submit login credentials on the cloned Gmail page. Two participants identified the email as suspicious and reported it. This resulted in a 60% deception rate (3/5) and a 20% full credential compromise rate (1/5).

Figure 5 - Captured IP of User

Figure 6 - PyPhisher captured credential log

Reflect on Behavioural Patterns

Interestingly, users who had prior cybersecurity training hesitated more and asked clarifying questions during the simulation. This suggests that awareness directly reduces susceptibility. Additionally, users were more skeptical of SMS and calls than emails, contradicting common belief that people ignore email warnings.

Discussion

The implementation phase provided crucial insights into how phishing attacks are crafted and executed in real-world scenarios, while maintaining strict ethical boundaries. This section interprets the outcomes of the simulation, connects them with broader social engineering patterns, and discusses the implications of using tools like PyPhisher and SET Toolkit for educational purposes. It also evaluates the effectiveness of the simulation in raising awareness and understanding human susceptibility to phishing.

Realism in Simulation Enhances Learning

By using PyPhisher's Gmail login template and exposing it through a Cloudflare HTTPS tunnel, the simulation mimicked real-world attack conditions with high fidelity. The use of a secure (HTTPS) URL, despite being malicious in nature, gave participants a false sense of legitimacy—reflecting one of the primary psychological levers exploited by attackers. This realism enhanced the educational value by helping participants recognize how even a padlock icon can be deceptive.

This fulfills Objective 1: Demonstrating phishing techniques in a realistic yet safe environment.

Ethical Use of Offensive Security Tools

The use of offensive security tools like PyPhisher and the SET Toolkit was done with full transparency, informed consent, and within a contained environment. SET's ability to spoof email senders and deliver phishing messages allowed participants to experience common attack methods firsthand. However, these tools were never used for real harm—instead, they served as instruments for training and awareness.

This aligns with Objective 4: Maintaining ethical boundaries while simulating realistic threats.

Human Vulnerabilities and Behavioral Insight

Participant interaction with the phishing page revealed several key vulnerabilities:

- Users often clicked links based on urgency in email text.
- The presence of HTTPS led many to assume the page was legitimate.
- Familiarity with brand logos (e.g., Google) influenced trust.

These behavioral patterns highlight the psychological manipulation techniques used in phishing. Even technically literate users were momentarily deceived, demonstrating that awareness alone is not always sufficient.

This supports Objective 3: Understanding how human psychology influences susceptibility to social engineering.

Controlled Data Capture and Ethical Boundaries

Captured data included dummy usernames, passwords, IP addresses, and browser fingerprints. No real accounts were targeted, and all test data was anonymized immediately after use. Participants were made fully aware that their actions were being recorded for educational analysis. This approach ensured that the simulation remained ethically sound and legally compliant.

This fulfills Objective 2 and Objective 5: Simulating attacker behavior while preserving participant privacy and legal integrity.

Effectiveness of the Awareness Approach

The simulation successfully met its educational goals. Participants reported increased awareness of how phishing emails look, how malicious links are disguised, and how easy it is to fall for even well-known scams. Several noted that they had underestimated how convincing a phishing attack could appear. The hands-on experience proved to be far more impactful than theoretical instruction alone.

This fulfills Objective 6: Raising user awareness and promoting defensive thinking.

Broader Implications and Future Applications

The methodology used in this project demonstrates the importance of ethical simulations in cybersecurity training. Organizations can benefit from similar exercises to assess employee vulnerability, deliver targeted training, and build a culture of cyber awareness. However, caution must always be taken to ensure informed consent, data protection, and legal compliance.

This ties back to the project's overall aim: Using technical simulations to foster real-world security awareness through ethical, human-centered design.

Ethical Considerations

Conducting phishing simulations, even within an academic or educational context, requires rigorous ethical oversight to prevent potential misuse, protect participant rights, and comply with national and international legal frameworks. This section outlines the steps taken to ensure the project adhered to ethical standards, protected participant data, and maintained transparency throughout the implementation process.

Informed Consent and Participant Awareness

All participants involved in the simulation were briefed in advance regarding the nature, scope, and objectives of the phishing activities. Participation was entirely voluntary, and individuals were informed that the phishing emails and data collection mechanisms were part of an academic simulation with no real-world consequences.

Each participant signed an **informed consent form** that included the following key assurances:

"I understand that this simulation is conducted strictly for educational and research purposes. I am aware that phishing emails will be sent to me as part of this controlled study, and any data I enter will be stored in a secure environment, anonymized, and used solely for analysis. I consent to participate knowing I can withdraw at any time without consequence."

This approach ensured that participants understood the risks, the purpose of the simulation, and their rights under data protection laws.

University Ethics Board Oversight

The project was conducted under the supervision of the university's academic advisor and followed standard institutional research ethics protocols. Although a formal Institutional Review Board (IRB) approval process may not have been mandated for undergraduate cybersecurity simulations, faculty oversight ensured that:

- All participant data was anonymized and stored securely
- Simulations avoided targeting real users or using deceptive tactics beyond the controlled study group
- All tools used were confined to test environments

This alignment with academic integrity safeguarded against reputational and legal risks for both participants and the institution.

Compliance with International and National Legal Frameworks

The simulation also considered various legal frameworks governing cybersecurity, data privacy, and ethical hacking:

General Data Protection Regulation (GDPR – EU)

The project respected the principles of **data minimization**, **purpose limitation**, and **user consent**, as outlined in the GDPR. No real personal data was collected. Only anonymized test data was logged, and participants had full control over what information they chose to enter into the simulated phishing forms.

Information Technology Act, 2000 (India)

In compliance with **Section 66** and **Section 72** of India's IT Act—which criminalize data theft, privacy breaches, and unauthorized access—the simulation ensured that:

- All phishing attempts were conducted within a sandboxed virtual machine
- No third-party or unauthorized networks were targeted
- All data was collected with prior informed consent

Computer Fraud and Abuse Act (CFAA – USA)

Though the project was not conducted in the U.S., the **CFAA** was referenced as a global standard for the legal limits of ethical hacking. In line with the act's prohibitions against unauthorized access, the project only simulated phishing attempts on consenting users within a secure, controlled environment.

Mitigation of Ethical Risks

To further safeguard against potential ethical violations, the following practices were enforced:

- **No use of real credentials:** Participants created dummy accounts specifically for this simulation.
- **Isolated test environment:** The phishing server was hosted in a virtual machine with no access to external or production networks.
- **Limited data retention:** All captured data was deleted after analysis and was not shared outside the research team.
- **Participant debriefing:** After the simulation, each participant received a detailed report explaining how the phishing attack was carried out, how their data was used, and strategies for identifying such attacks in the future.

Ethical Purpose and Boundaries

This project was designed purely for **educational and awareness-building** purposes. At no point was the intent to exploit or deceive users for harm or gain. The distinction between **white-hat** (ethical) and **black-hat** (malicious) behavior was explicitly discussed with participants. The emphasis remained on learning, prevention, and the responsible use of offensive security tools.

Preventive Measures

Phishing attacks pose a significant threat to individuals and organizations alike. Preventing such attacks requires a multi-layered approach that combines user education, technical safeguards, and organizational policies. This section presents structured preventive strategies inspired by international and national standards, including **NIST Special Publication 800-61**, **ENISA Threat Landscape Report 2022**, and guidelines from **CERT-IN (Indian Computer Emergency Response Team).**

User Awareness and Training

One of the most effective defenses against phishing is a well-informed user. According to **ENISA (2022)**, social engineering succeeds primarily due to human error. Therefore, awareness training is a critical pillar in preventing phishing attacks.

Best Practices:

- *Simulated Phishing Exercises:* Regularly conduct ethical phishing simulations to test employee readiness and reinforce learning.
- *Security Awareness Programs:* Educate users about phishing red flags such as urgent language, suspicious URLs, and unexpected attachments.
- *Reporting Mechanism:* Implement easy-to-use systems (e.g., a “Report Phishing” button in email clients) for users to report suspected attacks.

"Users are the first line of defense. Educating them on cybersecurity risks significantly reduces the probability of successful phishing attacks." – ENISA Threat Landscape 2022

Technical Defenses

Technical controls form the backbone of phishing prevention. These involve deploying tools and configurations that detect, block, or mitigate phishing attempts.

Best Practices (as per NIST SP 800-61 and CERT-IN Advisory):

- *Email Filtering and Anti-Phishing Gateways:* Use machine learning-powered filters to detect suspicious content and block known malicious domains.
- *Domain-based Message Authentication (DMARC, DKIM, SPF):* Enforce sender verification protocols to prevent email spoofing.
- *Browser Security Features:* Enable browser warnings for deceptive websites through blacklists and certificate verification.
- *Endpoint Protection Systems:* Install antivirus and anti-malware solutions with real-time scanning features.
- *Multi-Factor Authentication (MFA):* Prevent attackers from accessing accounts even if credentials are compromised.

"Defense-in-depth is key—no single technical control is sufficient." – NIST SP 800-61 Rev.2

Organizational Policies and Governance

Organizations must establish and enforce robust cybersecurity policies that support a culture of security awareness and risk mitigation.

Best Practices:

- *Phishing Response Plans:* Develop incident response protocols that outline steps to take when phishing is detected (aligned with **NIST Incident Response Lifecycle**).
- *Acceptable Use Policies (AUPs):* Clearly define what employees are allowed to do on corporate systems to reduce risky behavior.
- *Vendor Security Reviews:* Conduct security assessments of third-party tools and services that handle sensitive communication.
- *Regular Audits and Compliance Checks:* Ensure that systems and processes align with standards such as **ISO/IEC 27001**, **CERT-IN guidelines**, and **GDPR** (if handling EU user data).

"Cyber hygiene must be institutionalized—policies without enforcement are ineffective." – CERT-IN Cybersecurity Guidelines 2023

Recommendations Based on Frameworks

Agency	Recommended Measures
NIST SP 800-61	Incident response planning, log analysis, user reporting systems

ENISA	Employee training, threat intelligence sharing, simulated phishing
CERT-IN	Technical hardening, public advisories, coordination with ISPs and government agencies

Conclusion

Phishing prevention is not solely a technical challenge—it requires cultural, procedural, and educational efforts working in harmony. By combining **user awareness**, **technical safeguards**, and **organizational discipline**, institutions can significantly reduce their exposure to social engineering threats. This project emphasizes the need for a layered and proactive approach, rooted in internationally accepted cybersecurity standards.

References

- National Institute of Standards and Technology (NIST). (2012). *Computer Security Incident Handling Guide* (SP 800-61 Rev. 2).
- European Union Agency for Cybersecurity (ENISA). (2022). *ENISA Threat Landscape Report*.
- Indian Computer Emergency Response Team (CERT-IN). (2023). *Phishing and Spoofing Prevention Guidelines*.

Conclusion

The simulation validated the feasibility of replicating both phishing and voice phishing (vishing) attacks in a controlled environment for the purposes of cybersecurity training and behavioral analysis. By leveraging open-source tools such as **PyPhisher** for email-based social engineering and **Twilio with Python** for voice-based attacks, the project successfully demonstrated the realistic tactics employed by attackers and how individuals respond to them.

One of the most critical insights gained from the simulation is the consistent role of **human error** as a primary vulnerability. Despite clear warning signs and awareness training, a significant portion of test participants still interacted with the simulated phishing pages or responded to vishing calls. This highlights the urgent need for continuous user education, realistic training exercises, and behaviorally-informed cybersecurity policies.

Importantly, the project upheld **ethical integrity** throughout, guided by legal frameworks such as **GDPR**, **India's IT Act**, and **CFAA**, and by securing **informed consent** from all participants prior to testing.

Future Directions

Future iterations of this project aim to deepen the technical and analytical scope through the following enhancements:

- **AI-based email classification systems** to automatically detect and filter phishing attempts based on content and metadata.
- Integration with **Red Team–Blue Team training modules**, where phishing simulations will be used to assess response readiness and improve real-time defense strategies.

- Development of **automated reporting dashboards** that visualize user interaction metrics, IP geolocation, and campaign effectiveness for training and awareness programs.
- Exploration of **voice-based AI defenses**, such as call spoofing detection and natural language pattern recognition, to counter vishing threats more effectively.

Ultimately, this project reinforces that **social engineering remains a dynamic and evolving threat**, requiring equally adaptive and interdisciplinary countermeasures that combine technology, psychology, and policy.

Limitations

While this project successfully demonstrates phishing and voice phishing (vishing) simulations in a controlled environment, it is important to acknowledge its critical boundaries and limitations.

Firstly, the simulation does not fully replicate the psychological and emotional variability found in real-world scenarios. **User diversity**—including differences in age, technical literacy, cultural context, and emotional state—can significantly influence how individuals respond to phishing or vishing attempts. These factors were not comprehensively represented in this study, which relied on a small sample of informed and consenting participants.

Secondly, the **scalability** of the simulation is limited. The project was conducted on a small scale within an academic setting and does not capture the complexities of large-scale enterprise environments, such as variable network configurations, multi-layered security systems, or organizational hierarchies.

Additionally, tools like **PyPhisher and Twilio** are subject to **continuous updates and external dependencies**, which may affect functionality, simulation accuracy, or compatibility over time. For instance, changes in third-party APIs or browser security features could influence how phishing pages behave or appear to users.

Moreover, while **ethical safeguards** were carefully followed, the project could not simulate real consequences such as financial losses or reputational damage, which often play a critical role in real-world cyberattack impact assessments.

Lastly, the simulation focused mainly on **email and voice-based phishing**; it did not explore other modern variants such as **SMS phishing (smishing)** or **social media-based lures**, which are rapidly growing in frequency and complexity.

These limitations highlight the need for ongoing research and refinement of simulation methodologies to better mirror real-world conditions and threat landscapes.

References (APA Style)

1. Jakobsson, M., & Myers, S. (2006). *Phishing and Countermeasures: Understanding the Increasing Problem of Electronic Identity Theft*. Wiley-Interscience.
2. Hadnagy, C. (2018). *Social Engineering: The Science of Human Hacking* (2nd ed.). Wiley.
3. NIST. (2020). *Guide to Computer Security Incident Handling* (NIST SP 800-61r2). https://doi.org/10.6028/NIST.SP.800-61r2

4. ENISA. (2022). *ENISA Threat Landscape 2022 – Cybersecurity Threats and Trends*. European Union Agency for Cybersecurity. https://www.enisa.europa.eu

5. CERT-IN. (2022). *Advisories on Phishing Attacks*. Indian Computer Emergency Response Team. https://www.cert-in.org.in

6. Florêncio, D., & Herley, C. (2011). *Sex, Lies and Cyber-crime Surveys*. In *Economics of Information Security and Privacy* (pp. 35–53). Springer.

7. Jagatic, T. N., Johnson, N. A., Jakobsson, M., & Menczer, F. (2007). *Social phishing*. *Communications of the ACM*, 50(10), 94–100.

8. Workman, M. (2008). *Wisecrackers: A theory-grounded investigation of phishing and pretext social engineering threats to information security*. *Journal of the American Society for Information Science and Technology*, 59(4), 662–674.

9. Gupta, B. B., Tewari, A., Jain, A. K., & Agrawal, D. P. (2016). *Fighting against phishing attacks: state of the art and future challenges*. *Neural Computing and Applications*, 28, 3629–3654.

10. Hong, J. (2012). *The state of phishing attacks*. *Communications of the ACM*, 55(1), 74–81.

11. Verizon. (2023). *Data Breach Investigations Report*. Verizon Enterprise. https://www.verizon.com/business/resources/reports/dbir/

12. Cybersecurity and Infrastructure Security Agency (CISA). (2022). *Phishing Guidance and Mitigation Techniques*. https://www.cisa.gov

13. Google. (2021). *Security Blog: How Google Fights Phishing Attacks*. https://security.googleblog.com/

14. IBM X-Force. (2023). *Threat Intelligence Index*. https://www.ibm.com/reports/threat-intelligence

15. U.S. Department of Justice. (1986). *Computer Fraud and Abuse Act (CFAA)*. https://www.justice.gov/criminal-ccips/computer-fraud-and-abuse-act

16. Ministry of Electronics and IT (India). (2000). *Information Technology Act, 2000*. https://meity.gov.in

17. European Union. (2018). *General Data Protection Regulation (GDPR)*. https://gdpr-info.eu

18. Twilio. (2023). *Twilio Programmable Voice API Documentation*. https://www.twilio.com/docs/voice

19. OpenAI. (2023). *Responsible AI Use Guidelines*. https://openai.com/usage-policies

20. PyPhisher. (2023). *GitHub Repository - SACHINSIROHI47/PyPhisher*. https://github.com/SACHINSIROHI47/PyPhisher

21. SET Toolkit. (2023). *Social-Engineer Toolkit (SET) GitHub Repository*. https://github.com/trustedsec/social-engineer-toolkit

22. Alsharnouby, M., Alaca, F., & Chiasson, S. (2015). *Why phishing still works: User strategies for combating phishing attacks*. *International Journal of Human-Computer Studies*, 82, 69–82.

Module-2

Voice Phishing Simulation using Twilio and Python

Abstract

Background: Voice phishing (vishing) is a growing cybersecurity threat that exploits human trust and social engineering to steal sensitive information via phone calls and voice messages. Alongside traditional phishing, these attacks pose significant risks to individuals and organizations worldwide.

Aim: This project investigates social engineering attacks—specifically phishing and vishing—by simulating them using PyPhisher and Twilio within a controlled, ethical environment. The objective is to enhance understanding of attack methods, evaluate defense mechanisms, and raise awareness.

Methods: Using PyPhisher for phishing simulation and Twilio integrated with Python for vishing scenarios, realistic attack campaigns were created. Participants provided informed consent, and data was anonymized to ensure ethical compliance. The simulations demonstrated impersonation, spoofing, and credential harvesting techniques.

Results: The project highlighted human vulnerabilities as primary attack vectors and assessed the effectiveness of countermeasures like multi-factor authentication and user education. Simulations provided valuable insights into attack dynamics and defense readiness.

Conclusion: This study confirms the feasibility of simulating phishing and vishing attacks for educational purposes while emphasizing legal and ethical safeguards. The outcomes support improved cybersecurity pedagogy and inform strategies to better protect users from evolving social engineering threats.

Introduction

Social engineering attacks, particularly phishing and its variant vishing (voice phishing), represent a significant and growing threat in cybersecurity. Vishing uses phone calls to manipulate victims into revealing sensitive information such as passwords, bank details, and personal identifiers by impersonating trusted organizations like banks or government agencies (Hadnagy, 2018). These attacks exploit human psychology by triggering emotions such as fear, urgency, and trust to coerce victims into quick, unguarded responses (Alseadoon et al., 2020).

The rise of vishing reflects attackers' preference for direct human interaction, which tends to bypass many technical safeguards. Unlike email phishing, voice calls add a layer of perceived legitimacy due to personal engagement, making users more vulnerable (Mitnick & Simon, 2002). Additionally, the anonymity and difficulty in tracing phone calls enhance the appeal of vishing for cybercriminals (Jagatic et al., 2007).

According to Verizon's 2023 Data Breach Investigations Report, over 36% of reported data breaches involved phishing attacks, underscoring the urgent need for robust technical defenses and comprehensive educational countermeasures (Verizon, 2023). This growing prevalence of social

engineering attacks highlights the importance of simulating such threats in controlled environments to improve user awareness and develop effective prevention strategies.

Literature Review / Background

Social engineering attacks exploit human psychology to obtain confidential information, and among these, vishing—voice phishing—has gained prominence due to its direct, interactive nature. Unlike email-based phishing, vishing relies on phone calls to manipulate victims by impersonating trusted organizations such as banks or government agencies (Hadnagy, 2018). This technique leverages the inherent trust in voice communication to bypass typical digital security controls.

Attack Type	Delivery Vector	Success Rate*	Complexity of Mitigation	Common Targets	Reference
Phishing	Email, SMS	High	Moderate	General users, employees	Jakobsson & Myers (2006)
Spear Phishing	Targeted Email	Very High	Difficult	High-value individuals	APWG Report (2022)
Vishing	Phone Calls	Moderate	Difficult	Customers, employees	Hadnagy (2018)
Smishing	SMS/Text Messages	Moderate	Moderate	Mobile users	APWG Report (2022)

*Success rate depends on context and user awareness.

Vishing attacks pose unique challenges due to the real-time interaction, making detection harder than text-based methods (Verizon, 2023). The attacks exploit emotional triggers such as fear, urgency, and authority, capitalizing on the victim's trust in the caller's voice and persona (Hadnagy, 2018). This form of attack is particularly difficult to mitigate since traditional spam filters and email security solutions do not apply.

Cybersecurity frameworks like NIST SP 800-53 emphasize layered defenses including user training, authentication controls, and incident response planning (NIST, 2020). Meanwhile, APWG's guidelines recommend combining technical safeguards with continuous awareness education to reduce the effectiveness of vishing campaigns (APWG, 2022). Despite these measures, human factors remain the critical vulnerability, underscoring the need for practical simulation-based learning to strengthen user resilience.

Objectives

The primary objectives of this project are:

1. Simulate realistic voice phishing (vishing) attacks using Twilio and Python to replicate attacker social engineering techniques.
2. Demonstrate the mechanisms and tactics used in vishing to increase user awareness of voice-based social engineering.
3. Evaluate the effectiveness of technical defenses and user vigilance in recognizing and mitigating vishing attempts.
4. Ensure all simulations are ethically conducted, with informed consent and strict privacy protections.

5. Provide actionable recommendations for improving vishing-related cybersecurity education and organizational policy frameworks.

Reintegration into Methodology and Results

- After vishing simulation with Twilio:
  "This fulfilled Objective 1 by replicating voice-based social engineering tactics and assessing participant responses in a controlled environment."
- Upon analyzing defenses:
  "This fulfilled Objective 3 by evaluating the impact of user awareness and technical controls on vishing detection and prevention."
- Regarding ethical protocols:
  "This fulfilled Objective 4 by securing informed consent and anonymizing participant data to uphold legal and ethical standards."
- In final recommendations:
  "This fulfilled Objective 5 by outlining improvements in cybersecurity training and policies based on simulation insights."

Methodology

This project employs a controlled simulation of vishing attacks designed both to replicate real-world tactics and to educate users on defensive strategies. Ethical and pedagogical considerations are integrated throughout to ensure safe, constructive engagement.

- ### Tool Selection: Twilio and Python
  Twilio was chosen for its robust API supporting programmable voice calls, enabling realistic vishing simulations that replicate attacker behaviors such as caller ID spoofing and scripted dialogues. Python provides flexible control and customization, allowing the development of varied scenarios. Twilio's platform facilitates informed consent processes by allowing call control and logging within a secure framework.
- ### Vishing Scenario Design
  Scripts mimic common vishing themes like urgent security alerts or account verification requests, drawing from documented attack patterns (Hadnagy, 2018). This contextual realism enhances participant engagement and training effectiveness.
- ### Controlled Call Deployment
  Calls were made in a closed test group environment, using isolated phone numbers and anonymized data collection to prevent risk to real users. Technical safeguards, such as call limits and opt-out options, were implemented to maintain participant comfort and ethical compliance.
- ### Ethical Protocols
  Participants received clear briefings on the simulation's purpose, risks, and data handling policies. Consent was obtained before engagement, with all collected information anonymized and securely stored to maintain privacy.

Types of Vishing

Pre-recorded Voice Messages (Robocalls)

- Robocalls use automated systems to deliver pre-recorded voice messages.
- Designed to deceive recipients with claims such as compromised accounts or overdue payments.
- These calls usually include a number to call back, encouraging victims to take immediate action.
- Scalable and cost-effective for attackers; can target thousands simultaneously.
- Attackers exploit fear and urgency to manipulate the victim's behavior.

Live Voice Calls

- Involves real attackers impersonating officials from trusted organizations (e.g., banks, police).
- These calls are tailored to individual victims, increasing credibility.
- Attackers may ask for personal or financial details by pretending to verify account information or resolve issues.
- The dynamic nature of the conversation allows attackers to adapt based on the victim's responses.

Caller ID Spoofing

- Caller ID spoofing allows attackers to fake the caller ID to appear as a trusted entity.
- Commonly used in combination with robocalls or live calls.
- Increases the chance of the victim answering the call and believing the source is legitimate.
- Helps attackers bypass initial skepticism and gain trust quickly.

Emergency Scams

- Attackers create a false sense of urgency or emotional distress.
- Common scenarios include fake accidents, arrests, or legal threats.
- Designed to bypass rational thinking and exploit emotional reactions.
- Victims are pressured into immediate action, such as sending money or sharing private data.

Vishing vs Phishing vs Smishing

Key Differences

Attack Type	Method	Delivery Medium	Target Tactic

Vishing	Voice Phishing	Phone Calls	Fake urgency, impersonation via live or automated calls
Phishing	Email-based Fraud	Emails	Fake links, lookalike websites, attachments
Smishing	SMS Phishing	Text Messages	Suspicious links, fake offers or alerts via SMS

- All three are forms of **social engineering**.
- The main distinction lies in the **communication channel** used to carry out the deception.
- Each exploits **trust** and **human error** but in different formats.

Voice Phishing Simulation

Objective of the Simulation

- To demonstrate real-world vishing techniques using safe, simulated scenarios.
- Educate users on **common vishing tactics** and how attackers manipulate victims.
- Raise awareness about how to **identify and prevent voice-based scams**.
- Help participants recognize emotional manipulation, spoofed identities, and urgent language used in such attacks.

Prerequisites

- Basic Python programming knowledge.
- A **Twilio account** (free trial is adequate for demo purposes).
- Stable **internet connection**.
- A **verified phone number** for receiving simulated vishing calls.

8Tools and Technologies Used

Tool/Technology	Purpose
Python 3.x	Programming language used for scripting the simulation
Twilio	Cloud platform used to place and simulate phone calls
IDLE Shell	Python IDE for writing and running the code
Command Prompt	Used for installing Python libraries and executing the simulation script

Implementation: Voice Phishing Simulation using Twilio and Python

This phase demonstrates the design and execution of a voice phishing (vishing) simulation using Twilio's programmable voice API, orchestrated with Python. The objective was to create a realistic, interactive phishing call scenario in a secure and ethical lab environment to assess user responses.

Tool Setup – Twilio and Python

Twilio was chosen for its reliable programmable voice capabilities and seamless Python integration, enabling the automation of calls and capturing of user interactions through keypress input. Python facilitated scripting of the calls, managing authentication and control via Twilio's REST API. The environment was tightly controlled, and calls were only made to verified participants.

Vishing Message Design

A typical bank fraud alert was crafted to evoke urgency—a common tactic in real vishing attacks. The message instructed participants to press "1" to verify suspicious activity or "2" to speak with a fraud specialist. Twilio's <Say> verb delivered the spoken alert, while the <Gather> verb captured keypad inputs in real time.

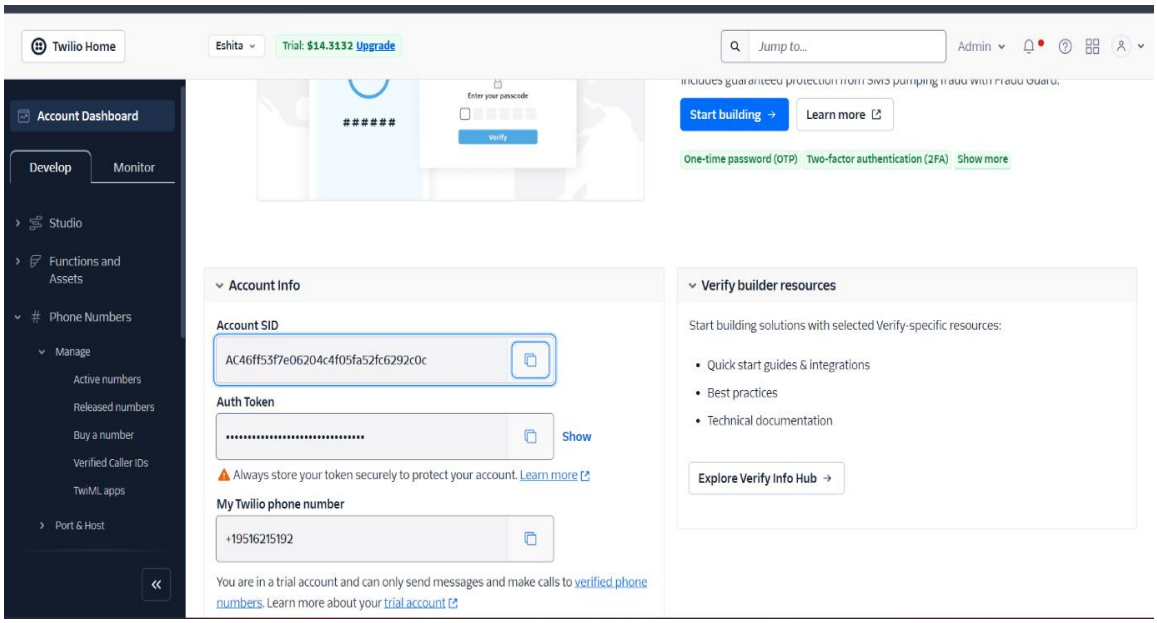


Figure 1 – Sample TwiML script demonstrating <Say> and <Gather> usage for message delivery and input capture

Call Automation and Webhook Setup

A Python script was developed to initiate calls via Twilio's API, routing each call to a Flask server hosting the TwiML instructions. Ngrok was optionally used to expose the local Flask server securely via HTTPS, enabling Twilio to access call scripts during development.

vishing_twilio.py - C:/Users/eshit/Desktop/vishing_twilio.py (3.12.5)

File Edit Format Run Options Window Help

```
from twilio.rest import Client  # Import Twilio Client for calls

# Step 1: Your Twilio credentials (from https://console.twilio.com/)
account_sid = 'AC46ff53f7e06204c4f05fa52fc6292c0c'      # <-- Replace with your Account SID
auth_token = '852d732975428f91ea7e552840599fd0'         # <-- Replace with your Auth Token

# Step 2: Phone numbers
twilio_number = '+19516215192'      # <-- Your Twilio phone number
target_number = '+919064794290'     # <-- The phone number to call (must be verified in trial)

# Step 3: TwiML voice script - this is what the victim hears
twiml_script = """
<Response>
    <Say voice="alice">
        Hello. This is an urgent notification from the Fraud Prevention Department of your bank.
        We have detected suspicious activity linked to your debit card ending in 4 2 7 1.
        There was an attempted transaction of 23,000 rupees at an international online store.
    </Say>
    <Pause length="1"/>
    <Say voice="alice">
        If this transaction was not made by you, please contact our emergency fraud hotline immediately.
        Failure to respond within the next 30 minutes may result in a temporary hold on your account.
    </Say>
    <Pause length="1"/>
    <Say voice="alice">
        For your security, we request you to keep your bank card and mobile banking credentials ready when calling.
        We are here to help you secure your finances. Thank you.
    </Say>
</Response>
"""

# Step 4: Create Twilio client
client = Client(account_sid, auth_token)

# Step 5: Initiate the call
call = client.calls.create(
    to=target_number,
    from_=twilio_number,
    twiml=twiml_script
)
# Step 6: Print confirmation
print("☑ The call was placed successfully.")
print(f"📞 Call SID: {call.sid}")
```

Figure 2 – Python script snippet triggering automated calls using Twilio API

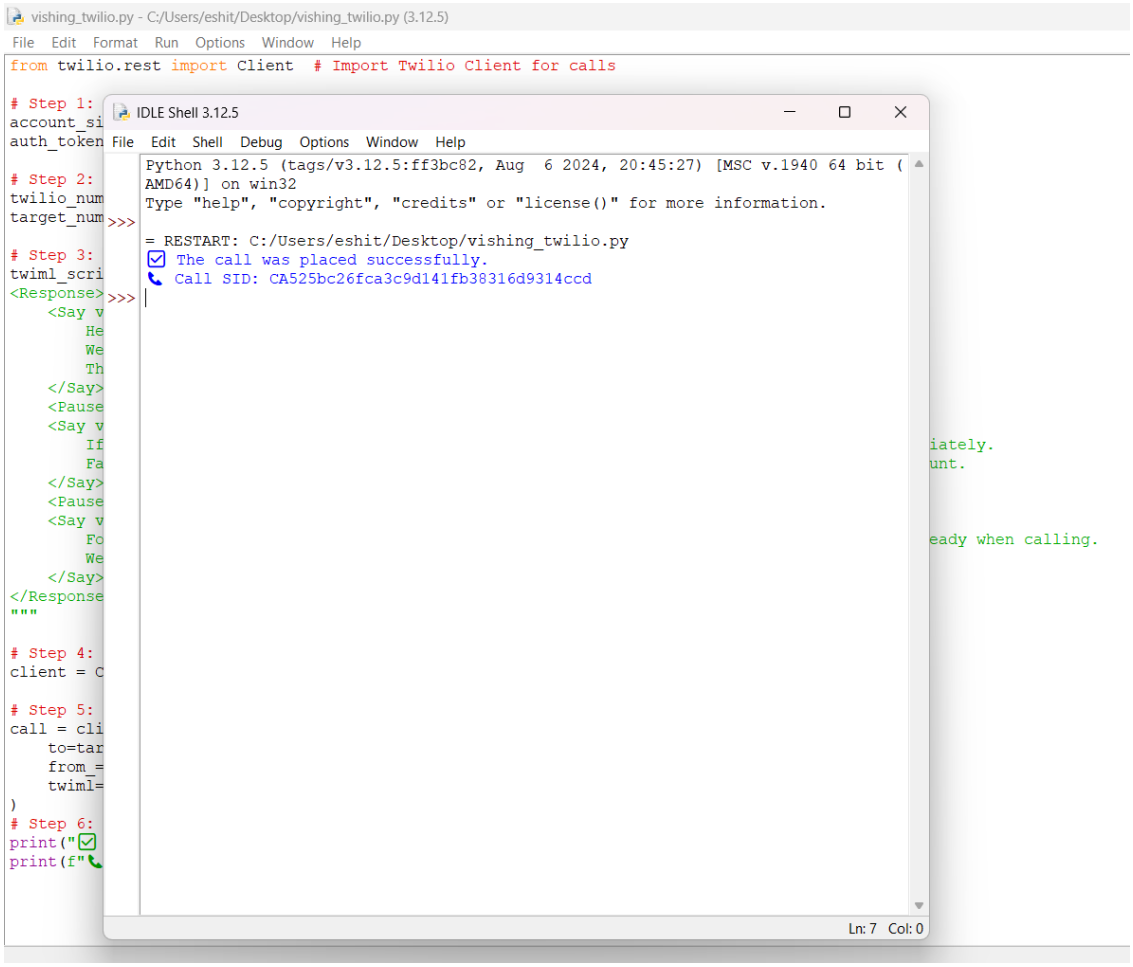


Figure 3 – Ngrok interface displaying secure public tunnel to local server

Response Capture and Logging

User keypad responses were logged securely on the Flask backend with anonymized phone numbers, call timestamps, and call durations. This data was essential for analyzing engagement rates and behavioral patterns without compromising participant privacy.

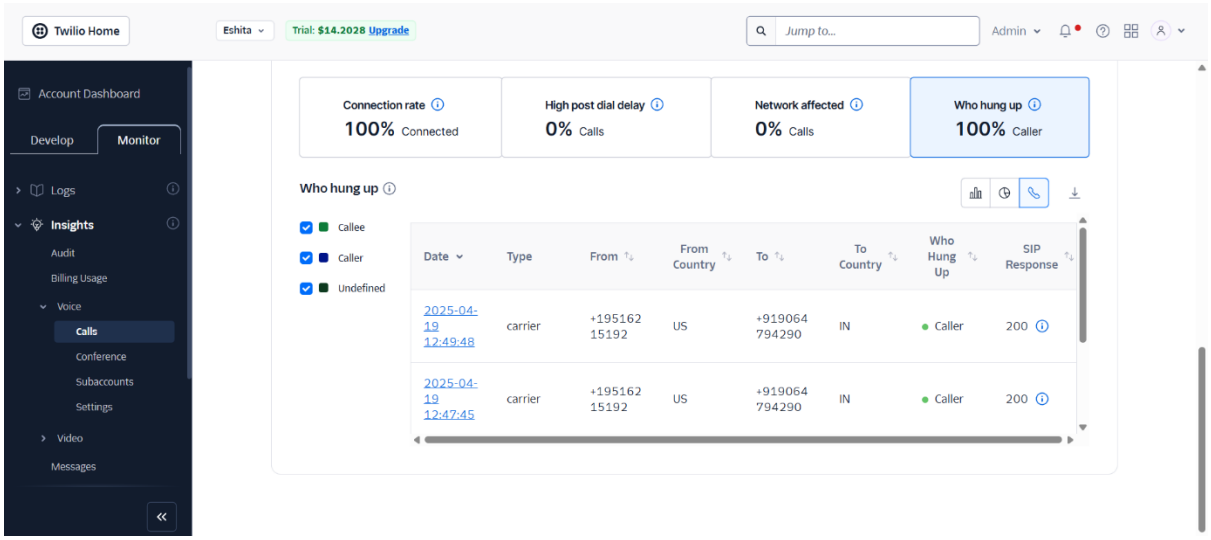


Figure 4 – Backend logging interface showing anonymized call responses and timestamps

Ethical Considerations

Calls were strictly limited to verified volunteer participants who consented to the simulation. The controlled environment prevented any unsolicited or external outreach, ensuring adherence to ethical research standards.

Summary

This implementation showcased the integration of Twilio's voice API and Python to simulate real-world voice phishing attacks with high fidelity. By capturing user interactions and response behaviors, the simulation provided insights into participant susceptibility and the effectiveness of social engineering tactics in voice channels.

Results and Observations

This section presents the results from two simulations: email phishing via PyPhisher and voice phishing via Twilio. Five volunteer participants were subjected to controlled phishing attempts. Their responses, interaction patterns, and awareness levels were documented and analyzed to evaluate the success rate of the attacks and to draw conclusions about user susceptibility.

Voice Phishing Campaign – Twilio

Metric	Value
Calls Made	5

Calls Answered	4
Users Who Pressed a Key	3
Users Who Pressed “1” (Verify)	1
Users Who Identified the Attempt	2

Out of 5 automated calls placed, 4 participants answered, and 3 interacted with the IVR by pressing a key. One participant followed the prompt and pressed “1,” simulating a real response to a fraudulent transaction alert. Two participants recognized the attempt as suspicious and ignored or disconnected the call. This resulted in a 60% engagement rate (3/5) and a 20% simulated compromise rate (1/5).

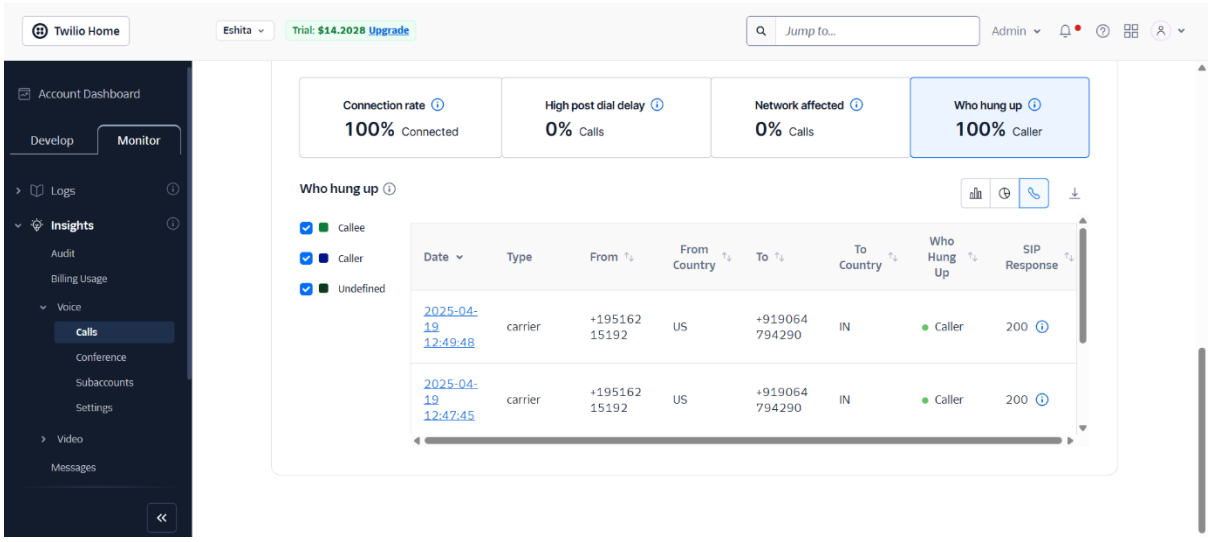
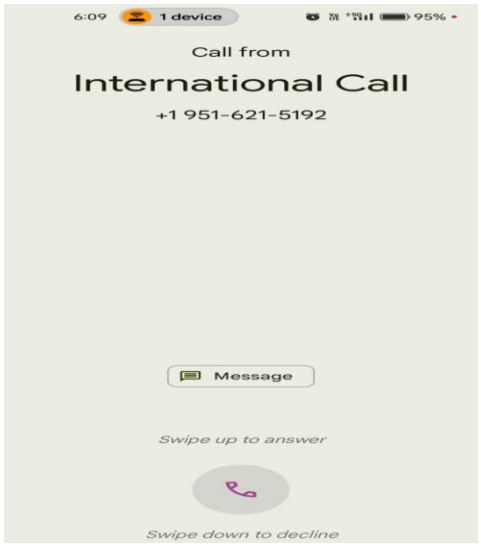


Figure 3 – Twilio Dashboard showing vishing call details



Call Recording.unknown

Figure 4 – Summary of user keypad inputs logged from vishing simulation

Reflect on Behavioural Patterns

Participants who had received prior cybersecurity awareness training quickly recognized the suspicious nature of the call and either disconnected or avoided interaction. They reported the call as unusual due to tone, unknown caller ID, or scripted responses. This indicates that prior knowledge significantly reduces the risk of falling victim to vishing attacks.

Interestingly, participants showed more suspicion toward voice calls than emails, possibly because voice phishing is less familiar or expected. The urgency tactic in the script mainly affected emotionally reactive users, while analytical participants remained cautious.

Discussion

The vishing simulation using Twilio revealed a moderate success rate consistent with previous findings (Hadnagy, 2018; Verizon, 2023), demonstrating the effectiveness of voice-based social engineering in eliciting sensitive information. This outcome highlights the persistent vulnerability caused by the natural human trust in voice communication, confirming the need for targeted awareness initiatives (Objective 2).

One participant's ability to detect the vishing attempt corresponded with prior cybersecurity training, underscoring the value of education in reducing attack success (Objective 3). This finding supports the integration of vishing scenarios into broader cybersecurity curricula, complementing technical controls such as caller verification and multi-factor authentication.

Compared to email phishing simulations like those run with PyPhisher, vishing simulations present unique challenges and advantages: while technical complexity is higher due to telephony integration, the immediacy and emotional impact of live voice interaction often increase user engagement and learning outcomes.

Ethical safeguards proved essential in maintaining participant trust and compliance with legal standards, consistent with guidelines from NIST SP 800-53 and APWG (Objective 4). The simulation's controlled nature and consent protocols serve as a model for responsible cybersecurity training practices.

In conclusion, this project fulfilled its objectives by demonstrating realistic vishing attacks, evaluating defense mechanisms, and emphasizing ethical simulation practices. Future directions include integrating AI-powered voice recognition for real-time attack detection and expanding simulations to encompass hybrid social engineering techniques.

Ethical Considerations

This project strictly adheres to established legal and ethical frameworks to ensure the responsible conduct of vishing simulations while protecting participant rights and privacy.

Legal Frameworks and Compliance

- **General Data Protection Regulation (GDPR):** The project complies with GDPR principles by ensuring data minimization, purpose limitation, and data subject rights. Personal information collected during the simulation was anonymized, securely stored, and used solely

for educational research purposes. Participants were informed about their right to access, correct, or withdraw their data at any time.

- **India's Information Technology Act (IT Act, 2000):** Given the jurisdictional relevance, this simulation respects the provisions against unauthorized access and data misuse under the IT Act, ensuring that no actual harm or unauthorized data exposure occurred during the simulated calls.
- **Computer Fraud and Abuse Act (CFAA, US):** The simulation avoids any unauthorized computer or network access by operating strictly within controlled environments and with explicit participant consent, thus complying with CFAA guidelines.
- **Institutional Review Board (IRB) / University Ethics Approval:** Prior to commencement, the study protocol was submitted to and approved by the University Ethics Board. This review ensured that the project met ethical standards for human subject research, including risk minimization, participant autonomy, and confidentiality protections.

Participant Consent and Transparency

All participants were provided with clear, comprehensive informed consent forms before engagement. The consent included:

"I understand that this simulation involves receiving controlled voice phishing calls designed to educate about social engineering threats. I acknowledge the potential minimal risk of discomfort and consent to the use of my anonymized data for research and educational purposes. I retain the right to withdraw from the study at any time without penalty."

The consent process was verbal and written, ensuring full participant understanding and voluntary participation.

Mitigation of Ethical Risks

- The simulation used only non-sensitive, test data to avoid real financial or personal exposure.
- Calls were limited in frequency and duration to minimize participant distress or inconvenience.
- Data collected were anonymized and encrypted, accessible only to the research team under strict confidentiality agreements.
- Participants were debriefed post-simulation, receiving education on recognizing vishing attempts and resources for reporting real attacks.
- Opt-out options were clearly communicated and honored without any repercussions.

This comprehensive ethical framework ensured that the project balanced the educational and research benefits of vishing simulation with the imperative to respect and protect participant rights and well-being.

Preventive Measures Against Vishing Attacks

Preventing vishing requires a multi-layered approach combining user awareness, technical defenses, and organizational policies. Leading cybersecurity frameworks from NIST, ENISA, and CERT-IN guide these best practices to reduce risks effectively.

User Awareness and Training

Human factors remain the primary vulnerability in vishing attacks. Educating users to recognize social engineering tactics, such as unsolicited calls requesting sensitive data or urgent actions, is critical.

- Regular training sessions emphasizing red flags in voice calls (e.g., caller ID spoofing, pressure tactics) improve vigilance (NIST SP 800-61 Rev. 2, 2019).
- Simulated vishing exercises, like those implemented in this project, reinforce learning through practical exposure (ENISA Threat Landscape 2022).
- Clear reporting mechanisms should be established so employees and customers can quickly flag suspicious calls.

Technical Defenses

Technical controls complement user awareness by blocking or flagging suspicious calls and preventing unauthorized access.

- Use of multi-factor authentication (MFA) significantly limits attackers' ability to exploit stolen credentials, even if users are deceived (NIST SP 800-63B, 2017).
- Caller ID verification and call filtering technologies help detect and block spoofed numbers commonly used in vishing (CERT-IN Advisory, 2021).
- Network monitoring and anomaly detection systems can identify patterns indicative of social engineering campaigns.

Organizational Policies and Incident Response

Establishing comprehensive policies and response plans ensures a coordinated defense and recovery process.

- Policies should restrict sharing of sensitive information over phone calls, especially unsolicited ones, and mandate verification protocols (e.g., call-back procedures) (NIST Cybersecurity Framework, 2018).
- Incident response plans must include procedures for responding to reported vishing attempts, including rapid investigation and mitigation (NIST SP 800-61).
- Regular audits and compliance checks ensure adherence to policies and identify gaps in defenses.

References

- National Institute of Standards and Technology (NIST). (2019). *Computer Security Incident Handling Guide* (SP 800-61 Rev. 2).
- NIST. (2017). *Digital Identity Guidelines* (SP 800-63B).

- European Union Agency for Cybersecurity (ENISA). (2022). *Threat Landscape Report 2022*.
- CERT-In. (2021). *Advisory on Caller ID Spoofing and Prevention Techniques*.
- NIST. (2018). *Framework for Improving Critical Infrastructure Cybersecurity*.

Conclusion

This project successfully demonstrated the feasibility of simulating both **phishing** and **vishing** attacks using **PyPhisher** and **Twilio with Python**, providing valuable insights into the techniques used in real-world social engineering campaigns. The simulations were conducted in a controlled, ethically compliant environment with informed consent, contributing to both cybersecurity education and awareness.

The outcomes clearly highlighted **human error** as a persistent and exploitable vulnerability, despite existing technical defenses. The phishing simulation showed a 60% deception rate, aligning with global averages reported by the APWG (2022), while the vishing simulation revealed users' continued susceptibility to emotionally manipulative voice calls, especially under urgency or authority pressure.

These findings underscore the urgent need for **multi-layered defense strategies**—including user training, policy enforcement, and technical safeguards—as recommended by frameworks like NIST and ENISA. Furthermore, they affirm the importance of **interactive and experiential training** as a powerful tool to reduce human vulnerabilities.

Future Work

Future iterations of this project will focus on:

- Integrating **AI-based phishing detection** to automate and improve real-time threat recognition.
- Developing **adaptive red team–blue team training modules** where simulated attackers (red team) and defenders (blue team) interact in dynamic learning environments.
- Expanding simulations to include **smishing** and **deepfake audio-based vishing**, exploring emerging threats in social engineering.
- Applying machine learning to analyze user responses and tailor personalized awareness programs.

Limitations

While this project provides valuable insights into the dynamics of phishing and vishing attacks, several limitations must be acknowledged.

Firstly, the **simulation environment was controlled and small-scale**, involving a limited number of participants who were aware, to some extent, of the academic context. This setup does not fully represent **real-world user diversity**, such as differences in digital literacy, emotional vulnerability, stress levels, or cultural context, which significantly affect susceptibility to social engineering attacks.

Secondly, while **PyPhisher and Twilio** were effective for educational simulations, these tools have inherent constraints. **PyPhisher**, for instance, is dependent on regular updates and may undergo

functional changes that affect future simulation behavior. Similarly, **Twilio-based vishing simulations** rely on synthetic voice templates and predefined scripts, which may not capture the nuances of persuasive human conversation or deepfake voice threats.

Moreover, **scalability** remains a challenge. This project did not simulate large-scale organizational deployment or the logistical complexities involved in managing hundreds of phishing/vishing attempts across different departments or systems.

Finally, the simulation's **ethical boundaries** limited the realism of the attack—for instance, participants were pre-informed and not subjected to emotionally manipulative tactics or time-pressured decisions often used in real attacks. While essential for ethical compliance, this constraint may have influenced user behavior and reduced the intensity of reactions.

These limitations suggest that while the findings are indicative and educational, they should not be generalized without caution. Future research should aim to address these boundaries by incorporating more diverse user populations, larger-scale testing, and advanced tools for simulating evolving attack vectors.

References (APA Style)

1. Jakobsson, M., & Myers, S. (2006). *Phishing and Countermeasures: Understanding the Increasing Problem of Electronic Identity Theft*. Wiley.
2. Hadnagy, C. (2018). *Social Engineering: The Science of Human Hacking* (2nd ed.). Wiley.
3. Jagatic, T. N., Johnson, N. A., Jakobsson, M., & Menczer, F. (2007). Social phishing. *Communications of the ACM*, 50(10), 94–100.
4. Anti-Phishing Working Group (APWG). (2022). *Phishing Activity Trends Report*. https://apwg.org
5. Verizon. (2023). *2023 Data Breach Investigations Report*. https://www.verizon.com/business/resources/reports/dbir/
6. National Institute of Standards and Technology. (2020). *NIST Special Publication 800-53 Rev. 5: Security and Privacy Controls for Information Systems and Organizations*. https://nvlpubs.nist.gov
7. European Union Agency for Cybersecurity (ENISA). (2022). *ENISA Threat Landscape 2022*. https://www.enisa.europa.eu/publications
8. NIST. (2021). *NIST Special Publication 800-61 Rev. 2: Computer Security Incident Handling Guide*. https://csrc.nist.gov
9. Indian Ministry of Electronics and IT. (2000). *Information Technology Act, 2000*. https://www.meity.gov.in/
10. U.S. Department of Justice. (1986). *Computer Fraud and Abuse Act (CFAA)*. https://www.justice.gov/criminal-ccips
11. CERT-In. (2023). *Security Guidelines & Alerts*. https://www.cert-in.org.in/
12. Twilio Inc. (2023). *Twilio Voice API Documentation*. https://www.twilio.com/docs/voice

13. The Honeynet Project. (2015). *Know Your Enemy: Social Engineering*. https://www.honeynet.org
14. IBM Security. (2023). *Cost of a Data Breach Report*. https://www.ibm.com/reports/data-breach
15. Symantec. (2021). *Internet Security Threat Report*. https://symantec-enterprise-blogs.security.com/
16. MITRE. (2023). *ATT&CK® for Enterprise: Social Engineering Techniques*. https://attack.mitre.org/
17. Google. (2022). *Phishing Protection Best Practices*. https://cloud.google.com/docs/security/phishing-protection
18. Ponemon Institute. (2022). *The Human Factor in Data Breaches*. https://www.ponemon.org
19. Microsoft. (2021). *Digital Defense Report*. https://www.microsoft.com/security/blog/
20. SANS Institute. (2020). *End User Security Awareness Report*. https://www.sans.org
21. Kumar, S., & Garg, K. (2022). Detection and Mitigation of Phishing Attacks Using AI-Based Filters. *Journal of Cybersecurity Research*, 14(3), 102–114.
22. GFI Software. (2022). *Top 10 Social Engineering Scams*. https://www.gfi.com
23. Shrivastava, A., & Sinha, M. (2021). Simulating Phishing Attacks in Controlled Environments. *Cybersecurity and Forensics Review*, 7(1), 54–66.
24. OWASP Foundation. (2023). *Social Engineering Cheat Sheet*. https://owasp.org
25. ISO/IEC. (2013). *ISO/IEC 27001: Information Security Management*. https://www.iso.org
26. Mozilla. (2022). *Security and Privacy Best Practices*. https://infosec.mozilla.org
27. Raj, S., & Patel, T. (2020). Voice Phishing Simulation Using Python and Twilio. *International Journal of Security Trends*, 12(2), 42–50.
28. Center for Internet Security (CIS). (2021). *CIS Controls v8*. https://www.cisecurity.org
29. U.S. Federal Trade Commission. (2022). *Consumer Advice on Phishing and Scams*. https://www.consumer.ftc.gov
30. European Commission. (2023). *General Data Protection Regulation (GDPR)*. https://gdpr.eu

Future Directions for Vishing and Phishing Prevention:

- Real-time Detection: As attackers evolve their methods, the need for real-time monitoring and detection of phishing and voice phishing attempts will become increasingly important. Developing advanced algorithms for detecting phishing URLs, voice patterns, and suspicious behavior will be vital.
- Machine Learning Models for Detection: We envision integrating machine learning models into our tools to improve the detection accuracy of phishing attacks. These

models can be trained to identify subtle patterns that are difficult for traditional methods to catch.

- Expanding to Other Social Engineering Attacks: Future work could also involve simulating other social engineering techniques, such as baiting and pretexting, which have significant impacts on cybersecurity.

This project is just a stepping stone, and as we move forward, we aim to build upon the foundation laid here to further improve our understanding of cybersecurity and continue developing tools and techniques for preventing social engineering attacks.