Navigating New Generation Phishing Campaigns: A Multi-Faceted Defense Against Sophisticated Cyber Threats

Urszula Piotrowska, Emil Merdzhanov, Cassandra Parker, Luis Sanjurjo

Project Objectives

1. Discuss emerging trends and best practices for user education and strengthening security awareness against phishing threats.

2. Demonstrate steps for email spoofing, secure configurations, and detecting/protecting against such attacks.

3. Enhance cybersecurity measures to protect individuals and organizations from social engineering phishing attacks.

Project Summary

1. **Project Introduction and Scope**
    - 1.1. Brief overview of the project objectives and focus on enhancing cybersecurity
    - 1.2. Threat Landscape Analysis
    - 1.3. Definition of Phishing
        - 1.3.1. Sophistication of Phishing
        - 1.3.2. Vulnerability of Gen Z to AI-Powered Phishing Attacks
        - 1.3.3. Phases of Modern Phishing Attacks

2. **Demonstration**
    - **2.1. Establishing a Domain and SMTP Server**
        - 2.1.1. Utilizing Brevo and GoDaddy to create a domain and set up email services
        - 2.1.2. Establish the Brevo SMTP server for email communication

    - **2.2. Creating a realistic Login Page**
        - 2.2.1. Demonstrating the process of creating a realistic login page to capture credentials convincingly

    - **2.3. Email Spoofing and Social Engineering Fusion**
        - 2.3.1. Utilizing Brevo for email spoofing and manipulation to simulate phishing attacks
        - 2.3.2. Showcasing the integration of social engineering tactics with email campaigns, exploiting human vulnerabilities
        - 2.3.3. Presenting a simple Python script for sending emails in Visual Studio Code

2.4. Deploying a Docker to Capture Credentials on a VM

2.4.1 Creating a docker to host a phishing webpage

2.4.2 Developing a script to capture credentials in human readable format

2.4.3 Demonstrating the process within a virtual machine (VM) environment

2.5. Deploying a Docker on a Amazon Lightsail Server

2.5.1 Transferring the Docker to Amazon Lightsail server using Filezilla

2.5.2 Configuring IP, firewall and port numbers

2.6. Conducting final testing to verify credential capture on the Amazon Lightsail server

3. Strengthening Defense: SPF, DKIM, DMARC, and Beyond

3.1. Implementing SPF, DKIM, and DMARC for email authentication and anti-spoofing measures

3.2. Exploring advanced techniques to enhance email security, including email encryption and digital signatures

3.3. Emphasizing the importance of proactive monitoring and continuous improvement to defend against evolving phishing threats

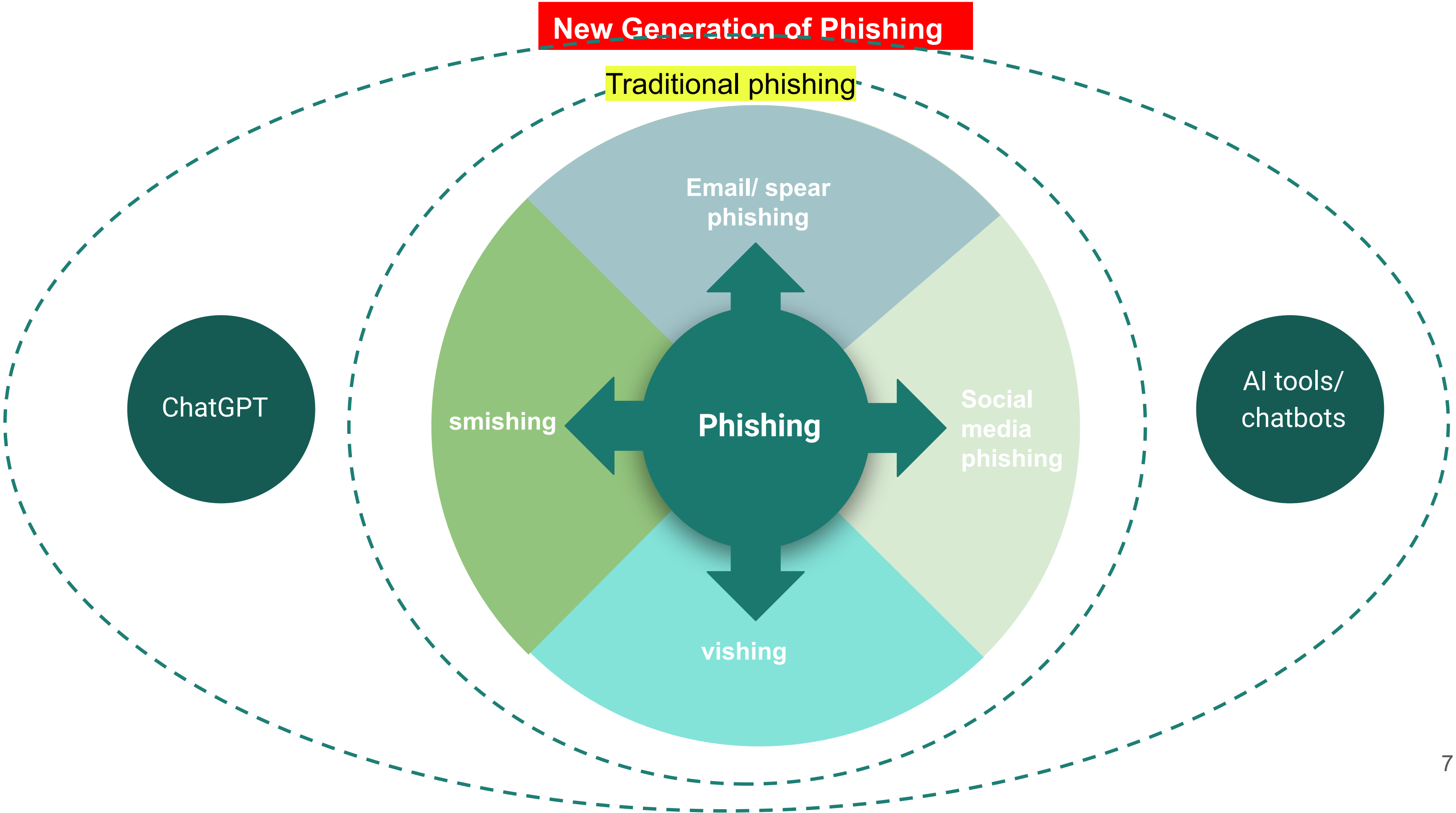
THREAT LANDSCAPE ANALYSIS 2022 / 2023

Definition of Phishing

A technique for attempting to acquire sensitive data, such as bank account numbers, through a fraudulent solicitation in email or on a web site, in which the perpetrator masquerades as a legitimate business or reputable person.

- *NIST, National Institute of Standards and Technology*

Sophistication of Phishing



Vulnerability of Gen Z to AI-Powered Phishing Attacks

01 **Communication Preferences**

- 90 percent of Gen Z are anxious about speaking on the phone*
- Text, apps, social media platforms, and chat-based interactions

02 **Digital Native Generation**

- Gen Z is tech-savvy and comfortable with digital platforms.
- Lacking human interactions

03 **Trust in Technology**

- Gen Z's trust in AI and technology may make them more vulnerable to phishing attacks
- Gen Z’s aren’t scared to share PII

04 **Less Experience with Traditional Phishing Techniques**

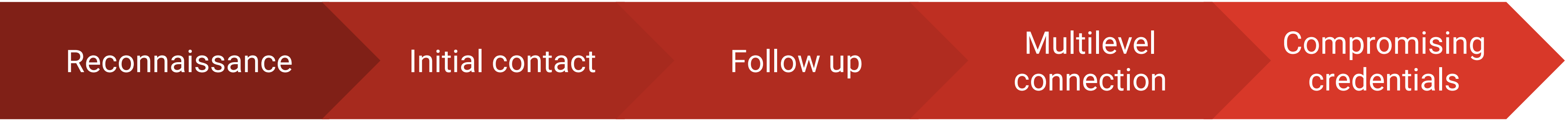
- Limited exposure to traditional phishing methods may leave Gen Z less cautious.

05 **Personalization and Targeting**

- Gen Z, accustomed to personalized content and targeted advertisements, may be more inclined to trust and engage with such tailored phishing attempts

*https://nypost.com/2023/07/10/gen-zs-aversion-to-phone-calls-has-created-a-brutal-dating-trend/

Phases of Modern Phishing Attacks



Reconnaissance

- Utilize AI-powered social media scrapers, data mining tools, and OSINT frameworks to swiftly gather a wealth of information, including emails, phone numbers, social media accounts, preferences, and more.

- Gathering as much data as possible about the target to create a personalized and convincing attack in relatively short time.

Initial contact

- AI algorithms by analyzing the target's online presence, preferences, and communication patterns crafting tailored content that increases the likelihood of the target responding or taking desired actions.

- ChatGPT tools can generate human-like responses, allowing attackers to simulate real-time conversations and adapt their messaging based on the target's interactions.

Follow up

- The attacker continues the conversation initiated to establish trust.

- AI and ChatGPT tools automate and enhance the interactions between the attacker and the target.

- AI generates responses across multiple channels like email, chatbots, and text messages simultaneously, scaling their operations and increasing the chances of success.

Multilevel connection

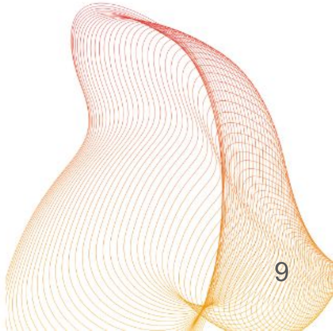
- AI and ChatGPT tools automate and orchestrate these multi-channel interactions seamlessly and efficiently.

- AI tools facilitate the generation of consistent and tailored messages across different platforms, enhancing the illusion of legitimacy and credibility.

- ChatGPT tools enable dynamic conversations, adapting responses based on the target's interactions, thereby increasing the effectiveness of the phishing attempt.

Compromising credentials

- The ultimate goal of the phishing attack is successfully obtaining the target's sensitive information.



DEMONSTRATION

Steps of the Demonstration

1. Establishing domain and SMTP server

2. Creating a realistic HTML Login Page

3. Developing a Python script to spoof email using Brevo SMTP and Visual Studio Code.

4. Initial credentials capture script testing on Docker in VM

5. Deploying container on Amazon Lightsail with configuration with GoDaddy

6. Final Testing on Amazon Server

1.

Establishing domain and SMTP server

CHALLENGES:

- Identifying an SMTP server capable of preserving the custom email header without overwriting it, thereby avoiding potential email spoofing

ACCOMPLISHMENTS:

- Identified a suitable SMTP server from Brevo that preserved the custom email header
- Established the domain and created the email address using GoDaddy, making communication within the domain smooth and efficient
- Obtained email services from Microsoft 365, providing us with a reliable platform to manage our emails.

TECHNOLOGY & TOOLS:

- Brevo, GoDaddy, Microsoft 365

2.

Creating a realistic HTML Login Page

CHALLENGES:

- Unable to copy the target HTML page using common cloning tools in Kali

ACCOMPLISHMENTS:

- Recreated the target page using HTML, CSS, and JavaScript with the help of developer tools in Google Chrome
- Created a highly realistic webpage

TECHNOLOGY & TOOLS:

- HTML, CSS and JS, Page Source, and Developer Tools in Google Chrome

3.

Developing a Python script to spoof an email using Brevo SMTP and Visual Studio Code.

CHALLENGES:

- Unable to utilize the known spoofing tools on Kali such as Espoofer and SET

ACCOMPLISHMENTS:

- Developed a simple Python script in Visual Studio Code for sending emails, that is able to send custom coded email header.
- Successfully sent an email and achieved email spoofing by obtaining the required SMTP server information.

TECHNOLOGY & TOOLS:

- Brevo STMP, GitHub, Python, Visual Studio Code

4.

Initial credentials capture script testing on Docker in VM

CHALLENGES:

- Creating a WebSocket service to return the user input from the “fake” URL to the back-end server and displaying it directly on the terminal proved challenging.
- Developing script that runs front & back end in one Docker Phishing application.
- Displaying the collected data in a human-readable format was another challenge.
- Although credentials were successfully captured, directing them to a text file remained **unresolved**

ACCOMPLISHMENTS:

- We were able to deploy the docker application and have it capture credentials from the index.html in the desired format using four scripts:

1. **Docker-compose.yml:** defines and sets up two services:
   a. **Webserver:**
      - deploys the website
      - It uses the httpd image, mounts the HTML, CSS, and JavaScript files
      - opens port 8080 for listening

   b. **Websocket-server:**
      - handles real time data
      - exposes port 8081 for communication between the user client and the terminal
2. **Script.js:**
   - handles the form submission event
   - collects credentials data, and sends it to the WebSocket server
3. **Dockerfile:**
   - builds the ‘websocket-server’
   - installs Node.js
   - sets the working directory
   - copies the server.js file
   - installs the 'ws' package
   - defines the command to run the server
4. **Server.js:**
   - the script that runs on the websocket-server
   - it’s responsible for printing that data captured by the website and on the terminal in human readable format.

TECHNOLOGY & TOOLS:

- JavaScript, .Yml, VM, GitHub, Google

5.

Deploying container on Amazon Lightsail with configuration with GoDaddy

CHALLENGES:

- Transferring the necessary files on Amazon Lightsail.
- Configuring Amazon firewall settings to allow proper incoming and outgoing traffic.
- Matching the script to the Amazon IP address and mapping out the ports to make it all work

ACCOMPLISHMENTS:

- Successfully added rules to the firewall settings to enable outgoing ports and listening ports.
- Created a new variable on the index.html to establish a WebSocket connection between front-end and back-end server

TECHNOLOGY & TOOLS:

- Amazon Lightsail server, GoDaddy, FileZilla, JavaScript

6.

Final Testing on Amazon Lightsail Server

CHALLENGES:

- Encountered difficulties in mapping certain ports as they differed from the ones on the local host

ACCOMPLISHMENTS:

- Successfully deployed the login page.
- Managed to capture the credentials.
- Resolved the majority of bugs during testing on the local machine

TECHNOLOGY & TOOLS:

- Amazon Lightsail server, JavaScript

Strengthening Defense: SPF, DKIM, DMARC, and Beyond

SPF

DNS-based method that validates the sender's domain by cross-referencing the IP addresses of the sending mail servers. When an email arrives, the recipient's mail server checks the SPF record to ensure the email originates from an authorized source, thus minimizing the risk of email spoofing.

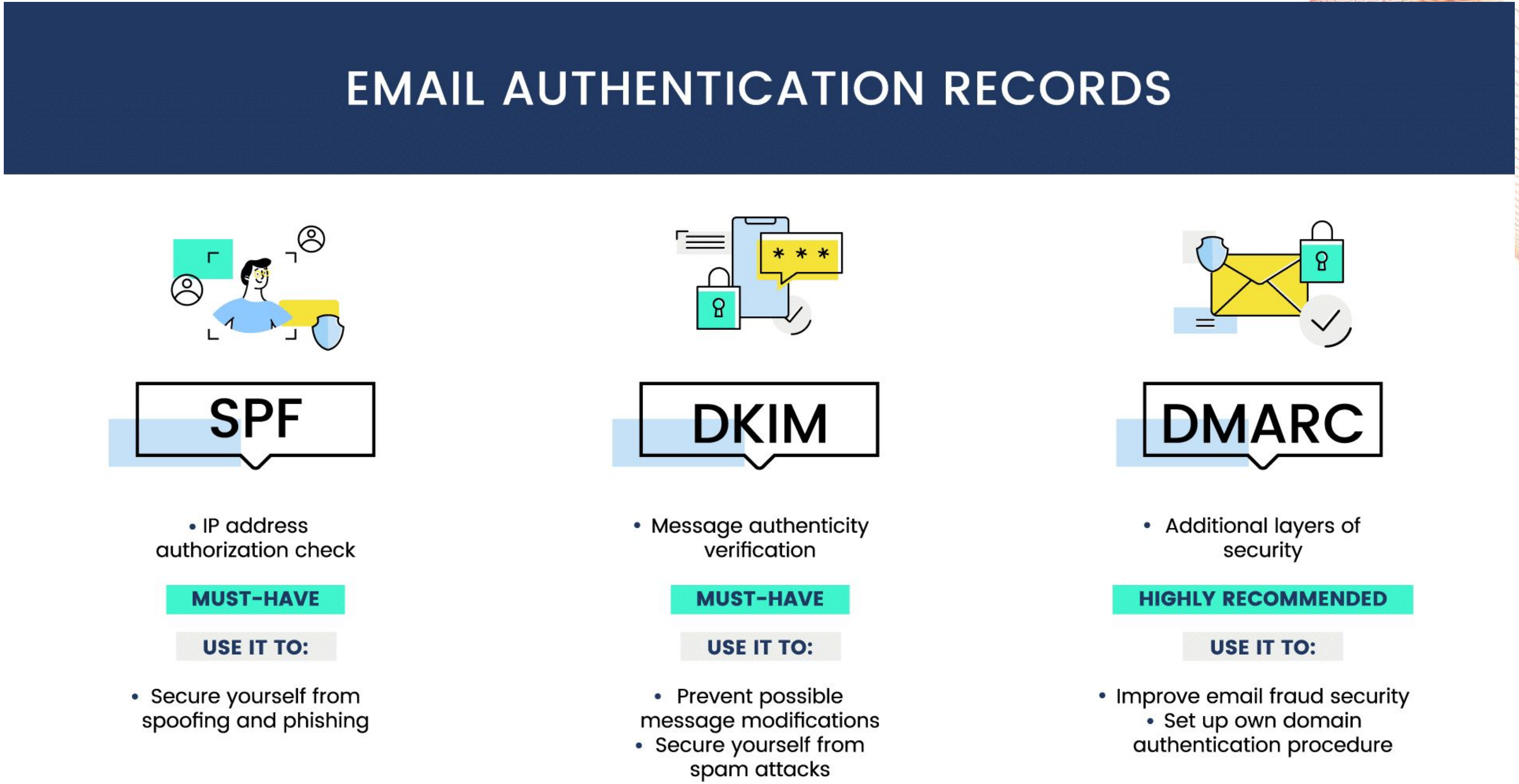
DKIM

adopts a digital signature-based approach to email authentication. It appends a cryptographic signature to outgoing emails from a domain. Upon receipt, the recipient's mail server verifies the DKIM signature by cross-referencing it with the public key stored in the domain's DNS record. DKIM ensures that the email content remains unaltered during transit and helps verify the email source.

DMARC

allows domain owners to instruct receiving mail servers on how to handle emails that fail SPF and DKIM authentication. DMARC offers flexibility in dealing with suspicious emails, such as quarantining or rejecting them, and provides reports on email authentication performance.

Strengthening Defense: SPF, DKIM, DMARC



EMAIL AUTHENTICATION RECORDS

SPF

- IP address authorization check

MUST-HAVE

USE IT TO:

- Secure yourself from spoofing and phishing

DKIM

- Message authenticity verification

MUST-HAVE

USE IT TO:

- Prevent possible message modifications
- Secure yourself from spam attacks

DMARC

- Additional layers of security

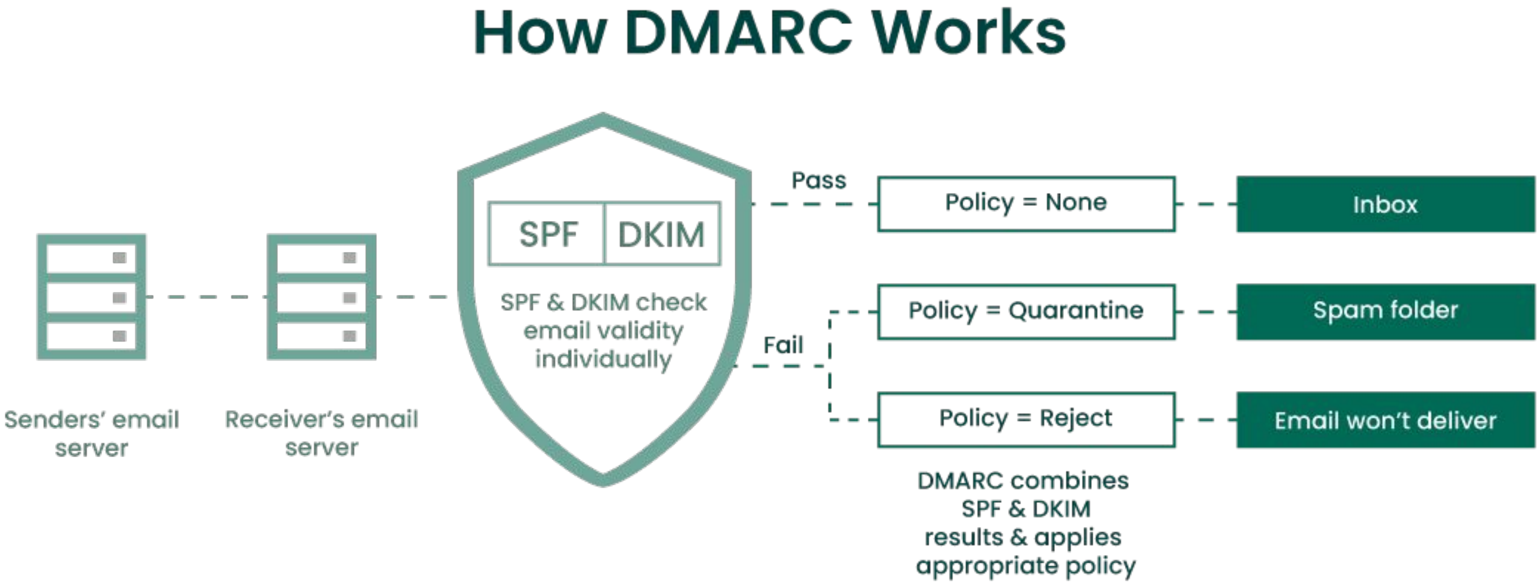
HIGHLY RECOMMENDED

USE IT TO:

- Improve email fraud security
- Set up own domain authentication procedure

Source: https://securityboulevard.com/2022/06/beginners-guide-to-dmarc-everything-you-need-to-know/

Strengthening Defense: SPF, DKIM, DMARC



How DMARC Works

Senders' email server

Receiver's email server

SPF DKIM

SPF & DKIM check email validity individually

Pass

Fail

Policy = None

Policy = Quarantine

Policy = Reject

Inbox

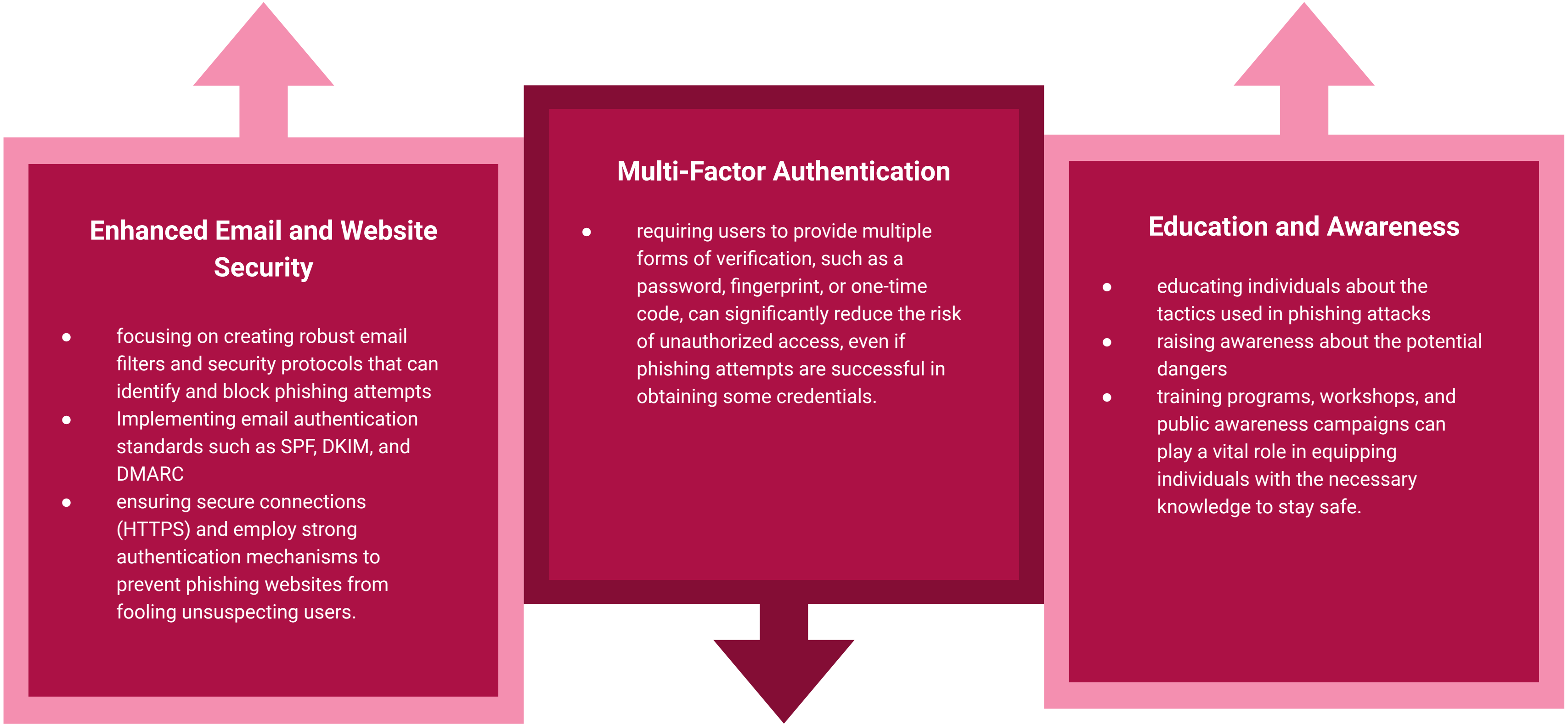
Spam folder

Email won't deliver

DMARC combines SPF & DKIM results & applies appropriate policy

Source: https://www.agari.com/solutions/email-security/dmarc

Mitigation



Enhanced Email and Website Security

- focusing on creating robust email filters and security protocols that can identify and block phishing attempts
- Implementing email authentication standards such as SPF, DKIM, and DMARC
- ensuring secure connections (HTTPS) and employ strong authentication mechanisms to prevent phishing websites from fooling unsuspecting users.

Multi-Factor Authentication

- requiring users to provide multiple forms of verification, such as a password, fingerprint, or one-time code, can significantly reduce the risk of unauthorized access, even if phishing attempts are successful in obtaining some credentials.

Education and Awareness

- educating individuals about the tactics used in phishing attacks
- raising awareness about the potential dangers
- training programs, workshops, and public awareness campaigns can play a vital role in equipping individuals with the necessary knowledge to stay safe.

Conclusion

1. Prioritize education on AI-powered phishing risks to raise awareness a among individuals and organizations.

2. Implement robust security measures, including advanced threat detection systems, to proactively defend against AI-driven phishing attacks.

3. Enhance cybersecurity measures to protect individuals and organizations from social engineering phishing attacks.