National Research Journal of Information Technology & Information Science
Volume No: 13, (January) Year: 2026 (Special Issue)
PP: 662-670
ISSN: 2350-1278
Peer Reviewed & Refereed Journal (IF: 7.9)
Journal Website www.nrjitis.in

PHISHING SIMULATION PLATFORM FOR ENHANCING CYBERSECURITY AWARENESS AND TRAINING

Abhishek Tiwari

Computer Science Engineering, Chandigarh University, **Mohali, Punjab**

Ayush Awasthi

Computer Science Engineering, Chandigarh University, Mohali, Punjab

Mukhtiar Singh

Computer Science Engineering, Chandigarh University, Mohali, Punjab

Nikhil Tripathi

Computer Science Engineering, Chandigarh University, Mohali, Punjab

ABSTRACT

Phishing attacks remain one of the most prevalent and damaging threats in the cybersecurity landscape. Educating users about phishing risks is crucial to mitigating these threats. This paper presents a phishing simulation platform designed to enhance cybersecurity awareness and train individuals in recognizing phishing tactics. The platform simulates realistic phishing scenarios by sending deceptive emails that redirect users to a phishing site, where credential entry is monitored. It comprises three core components: an email-sending module, a login page, and a real-time credential logging dashboard. The backend of the platform is developed using Flask, while the frontend is built with HTML, CSS, and JavaScript to ensure a responsive and interactive user experience. The system is deployed on Render, providing a scalable and accessible environment for real-time phishing simulations. By analyzing user interactions with simulated phishing attempts, the platform assesses users' ability to detect phishing threats. The results demonstrate its effectiveness in improving awareness and highlight its potential as a training tool for cybersecurity education. This study provides a foundation for future advancements in phishing awareness training and the development of more sophisticated educational tools to combat social engineering attacks.

Keywords: Phishing simulation, cybersecurity awareness, phishing attacks, email phishing, credential logging, phishing prevention, phishing education, cybersecurity tools, user behavior, real-time monitoring.

1. INTRODUCTION

Phishing attacks remain one of the most significant and prevalent threats in cybersecurity, often resulting in data breaches, financial losses, and reputational damage. Despite ongoing efforts to raise awareness and mitigate risks, phishing remains an effective technique used by cybercriminals to exploit human vulnerabilities. Consequently, understanding and combating phishing attacks has become a crucial focus in the field of cybersecurity education [1].

One of the most effective strategies to combat phishing is through awareness training, which helps individuals recognize phishing attempts and avoid falling victim to them [2]. Traditional training methods, such as workshops and static simulations, have proven beneficial but are often limited by their inability to adapt to rapidly evolving phishing tactics [3].

To address this, the introduction of dynamic phishing simulation platforms has gained traction. These platforms simulate real-world phishing scenarios, allowing individuals to experience firsthand the dangers posed by phishing attacks in a controlled environment [4]. However, while such platforms have been widely adopted, there remains a need for comprehensive, realistic, and interactive tools that can engage users more effectively and provide real-time feedback [5].

This research focuses on developing and evaluating a Phishing Simulation Platform for Spreading Awareness. The platform aims to enhance user education by providing a hands- on, realistic phishing simulation experience. By incorporating a variety of phishing techniques and real-time monitoring of user behavior, the platform seeks to improve users' ability to recognize phishing attempts and make more informed decisions when confronted with suspicious emails.

The primary parameters of this research include:

- Phishing Attack Simulation: The platform implements a phishing attack scenario where users receive a deceptive email containing a malicious link that leads to a phishing website designed to mimic a legitimate login page.

ISSN: 2350-1278
Peer Reviewed & Refereed Journal (IF: 7.9)
Journal Website www.nrjitis.in

- Email Sending Mechanism: A structured email-sending module is developed using smtplib in Python to deliver phishing emails with dynamically generated content and links.
- Phishing Website Development: The frontend is built using HTML, CSS, and JavaScript, while the backend, developed in Flask, captures and logs user credentials upon submission.
- Credential Logging and Storage: The platform successfully records any submitted credentials in a secure database, allowing for real-time monitoring of interactions with the phishing website.
- Deployment and Accessibility: The entire system is hosted on Render, ensuring remote accessibility and scalability. The integration between email dispatch, the phishing website, and credential logging functions seamlessly in a live environment.

The results of this study will contribute to the growing body of research on cybersecurity training tools and provide valuable insights into the development of more effective phishing simulation platforms

2. RELATED WORK

Phishing remains a persistent cybersecurity threat, and various research efforts have explored ways to mitigate its impact through simulation-based training and machine learning-driven detection systems.

Simulation-based phishing awareness programs have been widely studied, with findings indicating that repeated exposure to simulated attacks significantly improves user recognition of phishing attempts [6], [7]. A study on cybersecurity drill tests demonstrated that users, particularly executives, are highly susceptible to phishing emails, reinforcing the need for continuous training [6]. Another research effort analyzing email security frameworks highlighted the effectiveness of protocols like SPF, DKIM, and DMARC in mitigating email phishing but noted that human error remains a major risk factor [7].

Machine learning-based detection mechanisms have also gained traction in phishing prevention. Studies on Support Vector Machines (SVM) and Random Forest classifiers have shown that these models can detect phishing websites with high accuracy, often surpassing traditional rule-based detection systems [8]–[10]. Furthermore, hybrid approaches that combine multiple machine learning models have demonstrated improved detection rates, with Random Forest and Gradient Boosting ensembles achieving over 98% accuracy in identifying phishing websites [9].

While machine learning has significantly enhanced phishing detection, recent research on deep learning vulnerabilities has raised concerns regarding adversarial attacks. A study on CNN-LSTM models found that despite their high accuracy, these models remain susceptible to adversarial examples, emphasizing the need for robust defenses such as adversarial training [11].

Additionally, studies on intrusion detection systems (IDS) have examined the dual threat of phishing and denial-of- service (DoS) attacks, revealing that false alarms in IDS can amplify cyberattack damage. Simulation-based assessments suggest refining IDS models to reduce false positives while maintaining high detection accuracy [12].

These studies highlight the importance of combining simulation-based phishing awareness programs with AI-driven detection techniques to enhance cybersecurity. Our research builds on these findings by implementing a real-time phishing simulation platform that mimics credential-harvesting attacks and provides insights into phishing response mechanisms. Future advancements could integrate AI-based predictive analytics to personalize phishing awareness training and further reduce user susceptibility [10].

More such studies on different research on Phishing Simulation are shown in Table I.

3. METHODOLOGY

This section describes the technical approach used in developing the phishing simulation platform, which includes the email sending mechanism, phishing website setup, credential capturing system, and the real-time dashboard. The following steps outline the process:

3.1 Email Sending Mechanism

The phishing simulation begins with the generation and sending of phishing emails to users. These emails are crafted to appear as legitimate communications, with customized subject lines and content designed to deceive the recipient. The email contains a malicious link that redirects the user to the phishing website. The email sending process is automated using Python's 'smtplib' library, which allows for sending emails through an SMTP server. The email content and links are

ISSN: 2350-1278
Peer Reviewed & Refereed Journal (IF: 7.9)
Journal Website www.nrjitis.in

dynamically generated to simulate various phishing attack strategies, such as credential harvesting and malicious link redirects.

3.2 Phishing Website Setup

The phishing website is created to resemble a legitimate login page, prompting the user to enter sensitive information, such as login credentials. For this simulation, the website mimics common login forms from popular platforms. The website is set up using HTML and CSS for design, while JavaScript is used to simulate real-time interactions and data submission. The form submissions are captured and logged using Flask, a lightweight Python web framework, which is employed to create the backend server for handling user inputs and storing the submitted credentials.

3.3 Credential Capturing and Dashboard System

When users enter their credentials on the phishing website, the data is captured and stored in a secure database. The credentials are logged in real-time to assess the effectiveness of the phishing attack. To display the captured data, a web interface is used to create a real-time dashboard that shows the submitted credentials. This dashboard is designed to monitor the simulation progress, track user interaction, and visualize the collected data. JavaScript allows for the easy integration of interactive features and real-time updates, providing an intuitive user interface for both participants and administrators.

Frameworks and Tools Used:

- Python: The primary programming language used for the development of the phishing simulation platform. JavaScript: Used to create the user interface and real-time dashboard to monitor the phishing attack outcomes.
- Flask: Provides a backend server to handle user inputs from the phishing website and store the data. smtplib: A Python library used for sending phishing emails to users.
- HTML/CSS/JavaScript: Employed for the design and functionality of the phishing website.

Through this methodology, the platform simulates realistic phishing attacks and captures user behavior in real time, offering valuable insights into how individuals interact with phishing attempts and enhancing the effectiveness of awareness training.

Table 1. Summary of Phishing Email Response Literature Survey

Participants	Response Rates	Dataset	Key Findings	Year
~1,000 users over 4 years	Monthly variation; downward trend with training	Simulated phishing emails	Incentives had no impact; training reduced susceptibility	2018 [13]
305 employees (from 123 branches)	Opened: 11%, Clicked: 69% of opened, submitted: 34.3% of opened	Phishing emails + security questionnaire	Awareness: 79.3%. Weakest: Internet use (45.7%), Strongest: Password management (91.1%)	2019 [14]
39 employees	First test: 0% clicked, second: 31% clicked	Two phishing emails	Awareness high initially; second test tricked 31%	2019 [15]
21,000 users (20,495 employees, 697 executives)	Execs: Opened 3%, Clicked 12.3%, Submitted: 11.7% Employees: Opened 1.3%, Clicked 7%, Submitted: 15%	Whaling (Execs: iPhone 11 promo) Spear Phishing (Employees: Gmail storage)	27% of execs, 23% of employees opened; ~24% clicked links	2020 [16]

ISSN: 2350-1278
Peer Reviewed & Refereed Journal (IF: 7.9)
Journal Website www.nrjitis.in

6,000+ healthcare staff	General: 36% opened Custom: 62% opened	Three phishing simulations in an Italian hospital	Custom phishing more effective; fatigue increases risk	2022 [17]
31,940 participants	66% resisted phishing after 12 weeks of simulations	144 phishing simulations over a year	Most users resist phishing; ML predicts susceptibility; individualized training is key	2022 [18]
Multiple studies	Gamified training reduced phishing susceptibility by ~30%	N/A	Gamified training improves engagement and detection rates	2022 [19]
Employees of a large organization	Focus on click & reporting rates; ~50% neither click nor report	Real-world phishing + employee survey	Newer, unsatisfied employees are most vulnerable	2023 [20]
700 railway employees	First attack: 9.5%, Second: 8.0%, Both: 1.4%	Real-world phishing simulation	Training reduced response rate by 1.05%	2023 [21]
20,000 university participants	Credential Harvest: 24% click, 6.7% compromise Drive-by-URL: 46.7% click, 46.7% compromise Link in Attachment: 12.6% click, 1.5% compromise Link to Malware: 20.1% click, 9% compromise OAuth Grant: 26.5% click, 14% compromise	Tools: HiddenEye, Sendemail toolkit, Kali Linux	Drive-by URL most effective; cybersecurity awareness is crucial	2024 [22]

4. IMPLEMENTATION

This section outlines the steps involved in the development and deployment of the Phishing Simulation Platform for Spreading Awareness, covering the stages from initial implementation to version control and deployment.

4.1 Platform Setup and Framework Selection

The development of the phishing simulation platform began by selecting suitable technologies and frameworks. The decision was made to use Python due to its extensive libraries and ease of use for handling backend processes, including email sending and data logging. To build the real-time

dashboard and user interface, HTML, CSS, and JavaScript were chosen for their simplicity in creating interactive web applications. The Flask framework was employed to manage the backend and handle user interactions on the phishing website.

The project was divided into two main components: the email sending system and the phishing website with the credential logging mechanism.

4.2 Email Simulation Module

The email simulation module was created using the smtplib library in Python. This module automates the process of sending phishing emails to users as shown in Figure 1. The emails are designed to appear like legitimate communications, such as notifications from popular services (e.g., social media accounts, online banking, etc.). Each email contains

ISSN: 2350-1278
Peer Reviewed & Refereed Journal (IF: 7.9)
Journal Website www.nrjitis.in

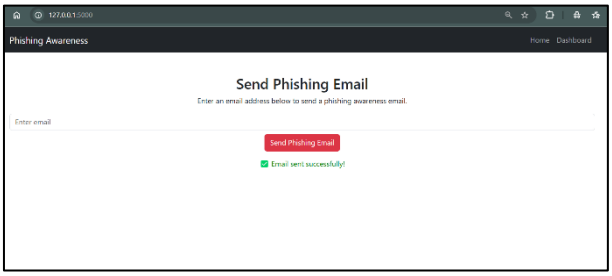

Fig. 1 Email Sending Page of the Phishing Simulation Website

a malicious link, which redirects the user to the phishing website. The email body, subject line, and link URL are dynamically generated to simulate different phishing tactics, such as credential harvesting and drive-by downloads, as shown in Figure 2.

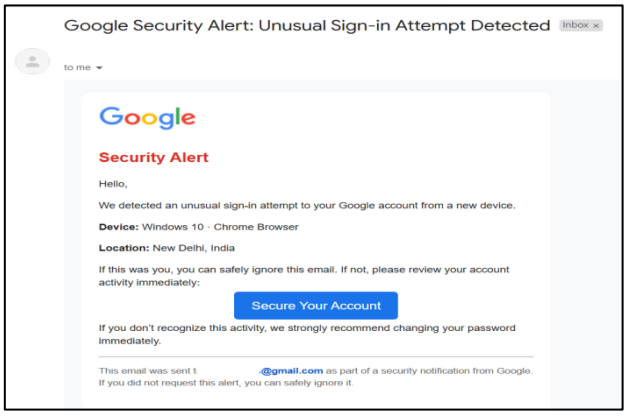

Fig. 2 Fake Email Message for Phishing

The system also logs the interaction with the email (whether the email was opened, if the link was clicked, and if the user entered any credentials on the phishing website). This data is stored for analysis in the next steps.

4.3 Phishing Website Development

The phishing website was designed to resemble real login pages from popular platforms. The website is built using HTML, CSS, and JavaScript to mimic a legitimate user interface. The page features a fake login form that prompts users to enter their credentials, such as username and password, as shown in Figure 3. The form submission is handled by a Flask server, which captures and stores the user inputs in a secure database.

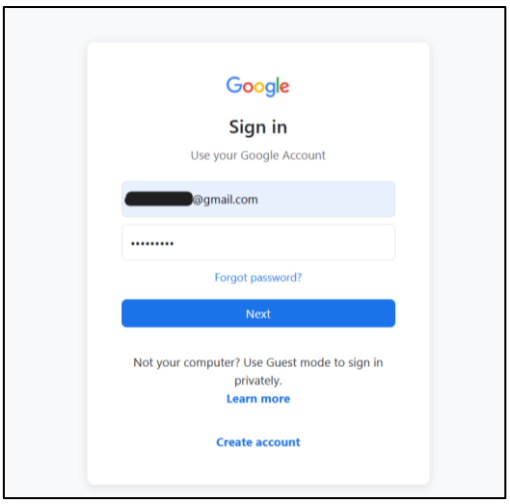

Fig. 3 Fake Login Page of the Phishing Simulation Website

ISSN: 2350-1278
Peer Reviewed & Refereed Journal (IF: 7.9)
Journal Website www.nrjitis.in

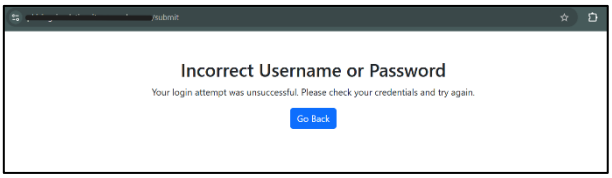

Fig. 4 Fake Failed Login Page

The website was hosted locally for testing, but it was later deployed on ”Render” for wider access in the final version. The use of JavaScript adds functionality such as form validation and real-time feedback, further enhancing the realism of the phishing attempt.

4.4 Credential Capturing and Dashboard

Once the user submits their credentials on the phishing website, the data is captured by the Flask backend and stored in an SQLite database, and the user is redirected to a fake failed login page as shown in Figure 4.

The credentials are then displayed in real-time on a PhishingSimulation site’s dashboard, where the administrator can monitor the submissions as shown in Figure 5. The dashboard visualizes the number of users who interacted with the phishing email, whether they submitted credentials, and provides insights into the effectiveness of the phishing attack.

Render was chosen for the website deployment platform because of its ease of use in integrating with Python and the ability to provide a seamless user experience. The dashboard is designed to update in real time as data is logged, giving administrators a clear view of the simulation’s progress.

4.5 Version Control and Collaboration

To ensure smooth development and easy collaboration, Git was used for version control. A GitHub repository was created to track the progress of the project and store all the source code as shown in Figure 6. This allowed the team to maintain a history of changes, collaborate on different components of the project, and ensure that all modifications were properly documented.

The development process was structured in the following way:

- Each feature (email sending, phishing website, dash- board) was implemented and tested separately.
- Branches were created for new features, and pull requests were used to merge them into the main branch once they were complete.
- Frequent commits ensured that the codebase was regularly updated and changes were well-documented.

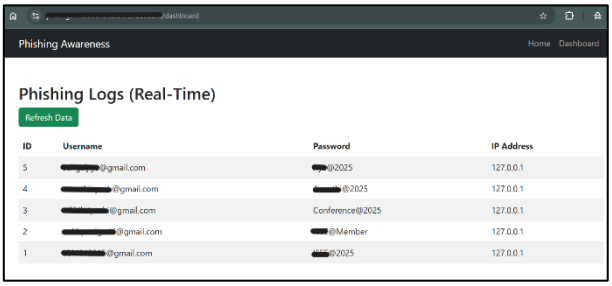

Fig. 5 Dashboard Page to real-time monitor user logs

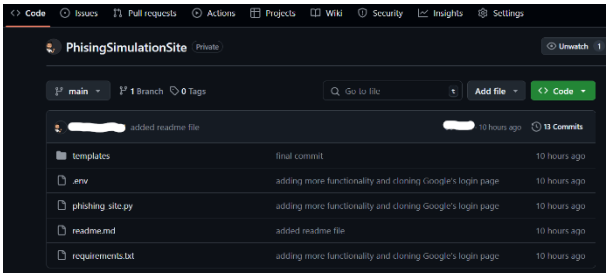

Fig. 6 GitHub Repository of the Phishing Simulation Site

ISSN: 2350-1278
Peer Reviewed & Refereed Journal (IF: 7.9)
Journal Website www.nrjitis.in

4.6 Deployment

Once the phishing simulation platform was fully implemented and tested locally, the project was deployed using Render for easy hosting as shown in Figure 7. The website was built using HTML, CSS, and JavaScript, while Flask handled the backend logic. A local database was used to store and display data on the dashboard.

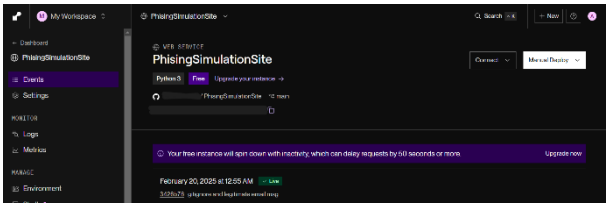

Fig. 7 Deployment of the Project on Render Platform

To ensure the platform's security, sensitive data (such as submitted credentials) was stored securely in the local database, and only authorized administrators had access to the dashboard for real-time monitoring.

5. RESULT AND DISCUSSION

The Phishing Simulation Platform for Spreading Aware- ness was successfully implemented and deployed, achieving all intended objectives. The project consists of three main components: an email-sending system, a phishing website, and a credential logging dashboard, all of which were integrated and tested to ensure smooth functionality.

5.1 Successful Implementation of Core Features

- Email Sending System: The email-sending module was developed using Python's smtplib library, allowing users to manually input an email address and send phishing emails with a deceptive link. The system was tested with different email providers to ensure proper delivery and reliability.
- Phishing Website: The frontend of the phishing website was developed using HTML, CSS, and JavaScript, mimicking a legitimate login page. The backend was implemented using Flask, enabling the capture and storage of user credentials upon submission. The site was hosted successfully and tested across different devices and browsers.
- Credential Logging Dashboard: The backend system stores submitted credentials in a secure database. A simple Flask-based dashboard was created for administrators to view and monitor captured credentials in real time.

5.2 System Testing and Validation

The platform was tested rigorously to validate the functionality of each component:

- Email Delivery Testing: The phishing emails were tested with different SMTP configurations to ensure successful delivery. Spam filter avoidance techniques were applied to prevent emails from being flagged.
- Phishing Website Testing: The website was tested for responsiveness and compatibility across multiple devices and browsers. The form submission mechanism was validated to ensure credentials were successfully captured and stored.
- Credential Logging Validation: The Flask backend correctly logged and displayed submitted credentials, ensuring seamless data transmission between the phishing website and the database.

5.3 Deployment and Accessibility

The platform was deployed using Render, ensuring scalability and ease of access. The deployment process included:

- Hosting the Flask backend and phishing website on Render, allowing users to access the simulation remotely.
- Configuring database storage to securely log captured credentials.
- Ensuring smooth operation and responsiveness across different network environments

6. CONCLUSION

The Phishing Simulation Platform for Spreading Aware- ness was successfully developed, tested, and deployed using Flask for the backend, HTML/CSS/JavaScript for the frontend, and Render for hosting. The platform effectively simulates real-

world phishing attacks by enabling email-based phishing attempts, capturing user credentials, and displaying them through a secure backend system. Through rigorous testing, all components functioned as intended, demonstrating the feasibility of a controlled phishing simulation for cybersecurity education. This project lays the groundwork for practical phishing awareness training, offering insights into how phishing attacks operate.

7. FUTURE SCOPE

Future improvements can enhance the platform's effective- ness and scalability:

- Automated Data Analysis – Implement analytics to track user interactions and measure phishing susceptibility.
- Expanded Phishing Attack Scenarios – Incorporate more attack types, such as spear phishing and malware- based phishing.
- User Awareness Training Module – Add educational content to help users recognize phishing tactics.
- Scalable Deployment – Optimize hosting and database storage for large-scale simulations.
- Machine Learning Integration – Use AI to detect phishing patterns and provide real-time phishing detection insights.

By addressing these areas, the phishing simulation platform could become an even more powerful tool for cybersecurity education and play a key role in mitigating the risks associated with phishing attacks

REFERENCES

1. F. Cremer, B. Sheehan, M. Fortmann, A. N. Kia, M. Mullins, F. Murphy, and S. Materne, "Cyber risk and cybersecurity: a systematic review of data availability," *Geneva Papers on Risk and Insurance - Issues and Practice*, vol. 47, no. 3, pp. 698–736, 2022.
2. B. Naqvi, K. Perova, A. Farooq, I. Makhdoom, S. Oyedeji, and J. Por- ras, "Mitigation strategies against the phishing attacks: A systematic literature review," *Computers & Security*, vol. 132, p. 103387, 2023.
3. I. Qabajeh, F. Thabtah, and F. Chiclana, "A recent review of conven- tional vs. automated cybersecurity anti-phishing techniques," *Computer Science Review*, vol. 29, pp. 44–55, 2018.
4. N. Marshall, D. Sturman, and J. C. Auton, "Exploring the evidence for email phishing training: A scoping review," *Computers & Security*, vol. 139, p. 103695, 2024.
5. D. Jampen, G. Gu¨r, T. Sutter, and B. Tellenbach, "Don't click: towards an effective anti-phishing training. a comparative literature review," *Human- centric Computing and Information Sciences*, vol. 10, no. 1, p. 33, 2020.
6. J. E. Lerums, L. D. Poe, and J. E. Dietz, "Simulation modeling cyber threats, risks, and prevention costs," in *2018 IEEE International Conference on Electro/Information Technology (EIT)*, pp. 0096–0101, 2018.
7. M. Kulkarni, Mohana, S. Kumar, M. Moharir, E. Baskaran, Y. Panjwani, and A. K. A. R, "Mitigating email phishing: Analytical framework, simulation models, and preventive measures," in *Proceedings of the 2024 10th International Conference on Communication and Signal Processing (ICCSP)*, pp. 1459–1464, IEEE, 2024.
8. S. Jain and C. Gupta, "A support vector machine learning technique for detection of phishing websites," in *2023 6th International Conference on Information Systems and Computer Networks (ISCON)*, pp. 1–6, 2023.
9. S. Kumar, G. P. Dubey, and B. Gupta, "Hybrid machine learning technique for prediction of phishing websites," in *2023 6th International Conference on Information Systems and Computer Networks (ISCON)*, pp. 1–4, 2023.
10. P. Bhatt, M. S. Obaidat, G. Dangwal, A. K. Das, M. Wazid, and
11. B. Sadoun, "Machine learning-based security mechanism for detecting phishing attacks," in *2024 International Conference on Communications, Computing, Cybersecurity, and Informatics (CCCI)*, pp. 1–6, 2024.
12. T. K. Yuji Ogawa and J. Cheng, "Vulnerability assessment for deep learning based phishing detection system," in *2021 IEEE Interna- tional Conference on Consumer Electronics-Taiwan (ICCE-TW)*, (Kyoto, Japan), pp. 1–13, IEEE, May 2021.
13. J. Shin, L. R. Carley, and K. M. Carley, "Simulation-based study on false alarms in intrusion detection systems for organizations facing dual phishing and dos attacks," in *Proc. of the 2024 Annual Simulation Conference (ANNSIM'24)*, (American University, DC, USA), pp. 1–13, Society for Modeling & Simulation International

ISSN: 2350-1278
Peer Reviewed & Refereed Journal (IF: 7.9)
Journal Website www.nrjitis.in

(SCS), May 20–23 2024.

14. S. McElwee, G. Murphy, and P. Shelton, "Influencing outcomes and behaviors in simulated phishing exercises," in *SoutheastCon 2018*, pp. 1– 6, 2018.
15. M. G. Ikhsan and K. Ramli, "Measuring the information security awareness level of government employees through phishing assessment," in *2019 34th International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC)*, pp. 1–4, 2019.
16. A. S. Abdullah and M. Mohd, "Spear phishing simulation in critical sec- tor: Telecommunication and defense sub-sector," in *2019 International Conference on Cybersecurity (ICoCSec)*, pp. 26–31, 2019.
17. S. Chatchalermpun, P. Wuttidittachotti, and T. Daengsi, "Cybersecurity drill test using phishing attack: A pilot study of a large financial services firm in thailand," in *2020 IEEE 10th Symposium on Computer Applications & Industrial Electronics (ISCAIE)*, pp. 283–286, 2020.
18. F. Rizzoni, S. Magalini, A. Casaroli, P. Mari, M. Dixon, and L. Coventry, "Phishing simulation exercise in a large hospital: A case study," *Digital Health*, vol. 8, pp. 1–13, 2022.
19. T. Sutter, A. S. Bozkir, B. Gehring, and P. Berlich, "Avoiding the hook: Influential factors of phishing awareness training on click-rates and a data-driven approach to predict email difficulty perception," *IEEE Access*, vol. 10, pp. 100540–100553, Oct. 2022.
20. N. Davis and E. S. Grant, "Simulated phishing training exercises versus gamified phishing education games," *2022 Fourth International Conference on Emerging Research in Electronics, Computer Science and Technology (ICERECT)*, pp. 1–8, 2022.
21. N. Beu, A. Jayatilaka, M. Zahedi, M. A. Babar, L. Hartley, W. Lewin- smith, and I. Baetu, "Falling for phishing attempts: An investigation of individual differences that are associated with behavior in a naturalistic phishing simulation," *Computers & Security*, vol. 131, p. 103313, Sept. 2023.
22. P. Sirawongphatsara, S. Prachayagringkai, P. Pornpongtechavanich, T. Rompun, K. Chaowmak, N. Phanthuna, and T. Daengsi, "Compar- ative phishing attack simulations: A case study of critical information infrastructure organization using two different contents," in *2023 10th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI)*, pp. 278–281, 2023.
23. A. Ciupe and B. Orza, "Reinforcing cybersecurity awareness through simulated phishing attacks: Findings from an hei case study," in *2024 IEEE Global Engineering Education Conference (EDUCON)*, (Cluj- Napoca, Romania), pp. 1459–1464, IEEE, April 21–24 2024.