Scientific Research Publishing

Journal of Software Engineering and Applications, 2024, 17, 522-529
https://www.scirp.org/journal/jsea
ISSN Online: 1945-3124
ISSN Print: 1945-3116

Enable Excel-Based Basic Cybersecurity Features for End Users by Using Python-Excel Integration

Mohamed Breik, Osama Magdy, Essam Amin, Tarek Aly, Mervat Gheith

Software Engineering Department, Faculty of Graduate Studies for Statistical Research, Cairo University, Cairo, Egypt
Email: mbreik@outlook.com, usamagdy@gmail.com, essam.amin@cu.edu.eg, Tarekmmmmt@pg.cu.edu.eg, Mervat_gheith@yahoo.com

How to cite this paper: Breik, M., Magdy, O., Amin, E., Aly, T. and Gheith, M. (2024) Enable Excel-Based Basic Cybersecurity Features for End Users by Using Python-Excel Integration. *Journal of Software Engineering and Applications*, **17**, 522-529.
https://doi.org/10.4236/jsea.2024.176029

Received: April 26, 2024
Accepted: June 21, 2024
Published: June 24, 2024

Abstract

In the digital age, the global character of the Internet has significantly improved our daily lives by providing access to large amounts of knowledge and allowing for seamless connections. However, this enormously interconnected world is not without its risks. Malicious URLs are a powerful menace, masquerading as legitimate links while holding the intent to hack computer systems or steal sensitive personal information. As the sophistication and frequency of cyberattacks increase, identifying bad URLs has emerged as a critical aspect of cybersecurity. This study presents a new approach that enables the average end-user to check URL safety using Microsoft Excel. Using the powerful VirusTotal API for URL inspections, this study creates an Excel add-in that integrates Python and Excel to deliver a seamless, user-friendly interface. Furthermore, the study improves Excel's capabilities by allowing users to encrypt and decrypt text communications directly in the spreadsheet. Users may easily encrypt their conversations by simply typing a key and the required text into predefined cells, enhancing their personal cybersecurity with a layer of cryptographic secrecy. This strategy democratizes access to advanced cybersecurity solutions, making attentive digital integrity a feature rather than a daunting burden.

Keywords

Python, End-User Approach, Excel, Excel Add-In, Cybersecurity, URL Check, API, Virustotal API, Encryption, Decryption, Vigenère Cipher, Python-Excel Integration

1. Introduction

The global character of the Internet has significantly improved our daily lives by

providing access to vast amounts of knowledge and enabling seamless connections. However, this interconnected world is fraught with risks, including malicious URLs that masquerade as legitimate links with the intent to compromise systems or steal sensitive information [1]. As cyberattacks become more sophisticated and frequent, identifying and mitigating these threats is crucial. Antivirus software has evolved to safeguard against harmful URLs, yet it remains out of reach for many non-technical users [2].

End-user development (EUD) is advancing to make technology accessible to both experts and non-experts, with tools that capture, represent, visualize, analyze, and test developer intent [3]. Integrating Python with Excel—a widely used tool in business and academia—offers a unique opportunity to democratize cybersecurity by providing advanced features in a familiar interface. This study aims to empower average end-users by integrating Python with Excel to deliver cybersecurity functionalities such as URL safety checks using the VirusTotal API and text encryption/decryption via the Vigenère Cipher.

2. Background

2.1. Computer Security

The field of computer security has evolved significantly since its early days in the 1970s. Ken Thompson's seminal work, "Reflections on Trusting Trust" (1984), highlighted the fundamental problem of trust in computer systems, emphasizing the need for secure infrastructure and tools [4]. This study builds on these foundational ideas by integrating security features into everyday tools like Excel.

2.2. Web Applications and Security Testing

Web applications are critical in today's digital landscape, serving as the backbone for communication, information exchange, and service provision. Security testing of these applications is vital to ensure their integrity and protect sensitive data [5]. Regular vulnerability assessments help organizations proactively address security flaws and comply with industry regulations.

2.3. The Vigenère Cipher

The Vigenère cipher, a method of encrypting text using a polyalphabetic substitution technique, has been a resilient cryptographic method for centuries. It employs a repeating key to shift letters in the plaintext, providing a simple yet effective means of securing communications [6]. This study leverages the Vigenère cipher to offer encryption and decryption functionalities within Excel, enhancing user data privacy.

3. Related Work

Several studies highlight the importance of large-scale analyses and classifications of malicious URLs. Choo *et al.* (2023) [7] conducted a comprehensive study on VirusTotal (VT) reports, providing insights into the characteristics and pat-

terns of malicious URLs. Shalaginov *et al.* (2016) [8] emphasized the role of large-scale datasets in malware detection research. The need for user-friendly cybersecurity tools that leverage such comprehensive data is evident. This study fills this gap by integrating advanced cybersecurity features into Excel, making them accessible to non-technical users.

End-User Development (EUD) security is a crucial area that aims to empower end-users to develop and adapt systems themselves while ensuring the security of these systems. Lieberman *et al.* (2006) [9] emphasize the goal of EUD in empowering end-users to develop and adapt systems themselves. Ko *et al.* (2011) [10] provide an overview of End-User Software Engineering (EUSE) and related terminology, which is essential for understanding the landscape of EUD security. Fischer *et al.* (2017) [11] propose EUD methods as a solution for developing flexible systems that can be adapted to different end-user needs directly. To enhance security in EUD environments, Rak *et al.* (2014) [12] focus on developing secure cloud applications, emphasizing the importance of mapping high-level security requirements with low-level interactions among application components.

In conclusion, EUD security is a multifaceted domain that requires a holistic approach integrating end-user empowerment with robust security measures. By leveraging HCI principles, user participation, and tailored security management systems, EUD environments can be fortified against potential threats, ensuring both usability and security for end-users.

4. Methodology

This study followed the Software Development Life Cycle (SDLC) to develop the Excel add-in, serving as proof of concept for enabling Excel-based cybersecurity features [13].

4.1. Plan and Requirement Analysis

• User Profiling: Define typical end-user profiles based on their cybersecurity knowledge and Excel usage patterns.

• Feasibility Study: Analyze the technical feasibility and limitations of integrating the VirusTotal API and encryption/decryption algorithms into an Excel add-in.

4.2. Design

• System Architecture: Outline the overall architecture of the Excel add-in, including the user interface, functional modules, and interaction with external APIs.

• VirusTotal API Integration: Design the integration process for the VirusTotal API, ensuring robust and secure communication between Excel and the VirusTotal service.

• Encryption/Decryption Module: Design the module for encryption and decryption, deciding on cryptographic algorithms that balance security with ease of use for non-technical users.

• As shown in Figure 1, the proposed system includes several key components (User, Excel Application, Data, and Local add-in server).

4.3. Implement

• Add-in Development: Utilize a suitable programming language (python, flask) to create the Excel add-in.

• URL Check Code Development: Prepare Virustotal API python code.

• Encrypt/Decrypt python Code Development: Prepare Vigenère Encryption and Decryption python code.

• Figure 2 shows the implemented architect.

4.4. Test

• Unit Testing: Perform unit tests on each module (API integration, encryption/decryption) to ensure they function correctly in isolation.

• Integration Testing: Test the integration of modules within the Excel environment to ensure they work together seamlessly.

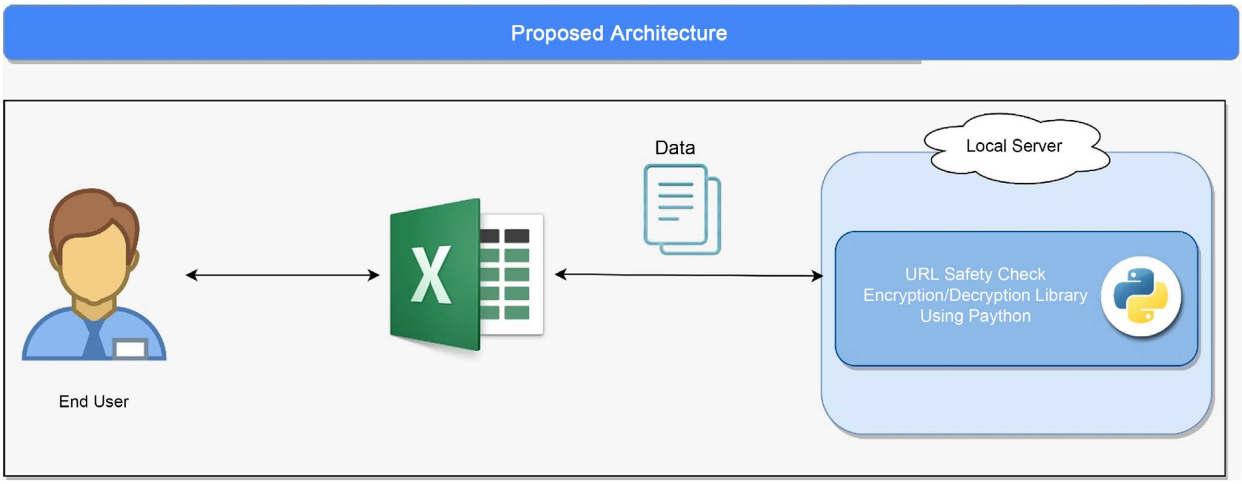

Figure 1. Proposed architecture.

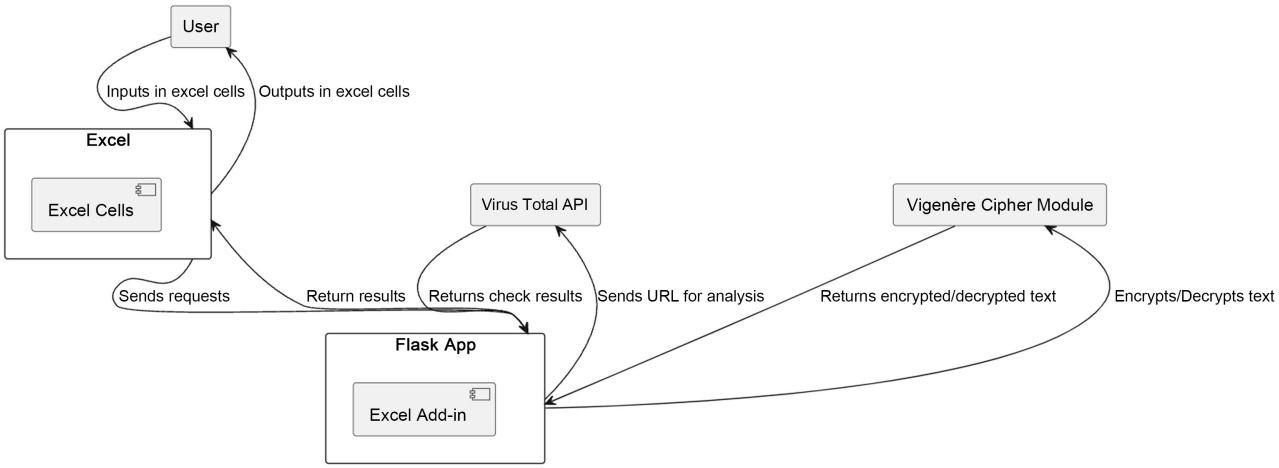

Figure 2. Implemented architecture.

4.5. Deploy

- URL Check API Deployment:

Embed the VirusTotal API into the add-in, ensuring that URLs can be checked seamlessly from within Excel as shown below in Figure 3.

1) User enter URL to check in cell A11
2) User click on URL Check add-in module
3) Results show up in cell B11

- Encryption/Decryption Deployment:

Embed the encryption/decryption functionality, allowing users to input a key and message within Excel cells to perform the cryptographic operations:

o Encryption as shown below in Figure 4:

1) User enters his Key in cell A14
2) User enters his text to encrypt in cell B14
3) User clicks on Encrypt add-in module
4) Encrypted text shows up in cell A15

o Decryption as shown below in Figure 5:

1) User enters his Key in cell A18
2) User enters his text to encrypt in cell B18
3) User clicks on Decrypt add-in module
4) Decrypted text shows up in cell A19

5. Contribution

This research makes several key contributions to the field of cybersecurity and end-user data protection:

5.1. Innovation in End-User Cybersecurity

• It introduces a novel Excel add-in that integrates with the VirusTotal API, bringing advanced URL safety checks directly into a commonly used application.

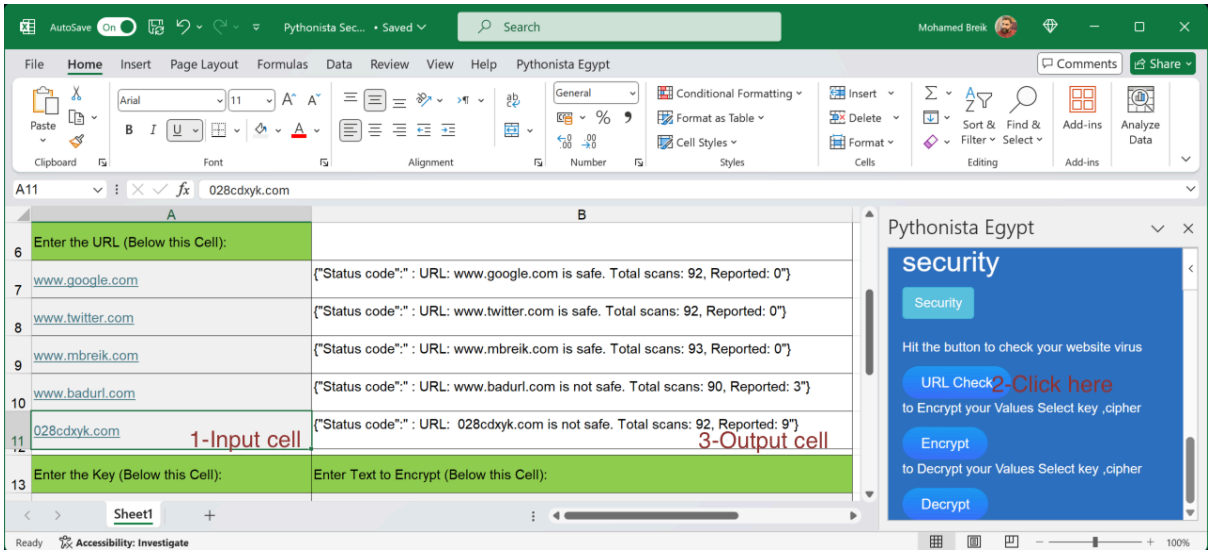

Figure 3. URL check Excel add-in.

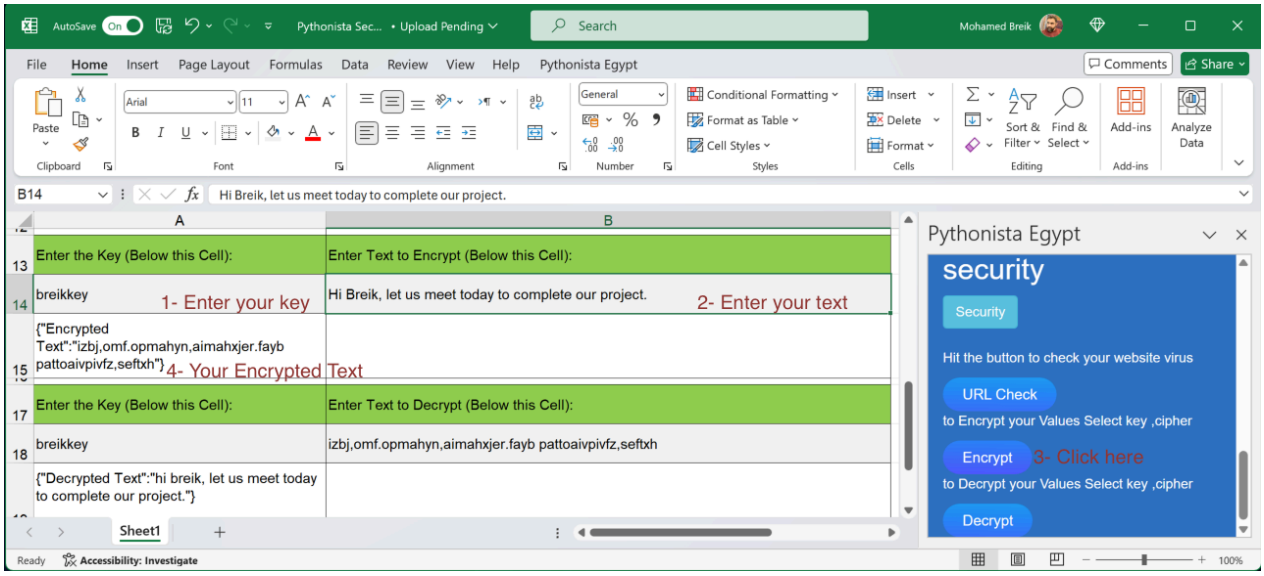

Figure 4. Encryption using Excel add-in.

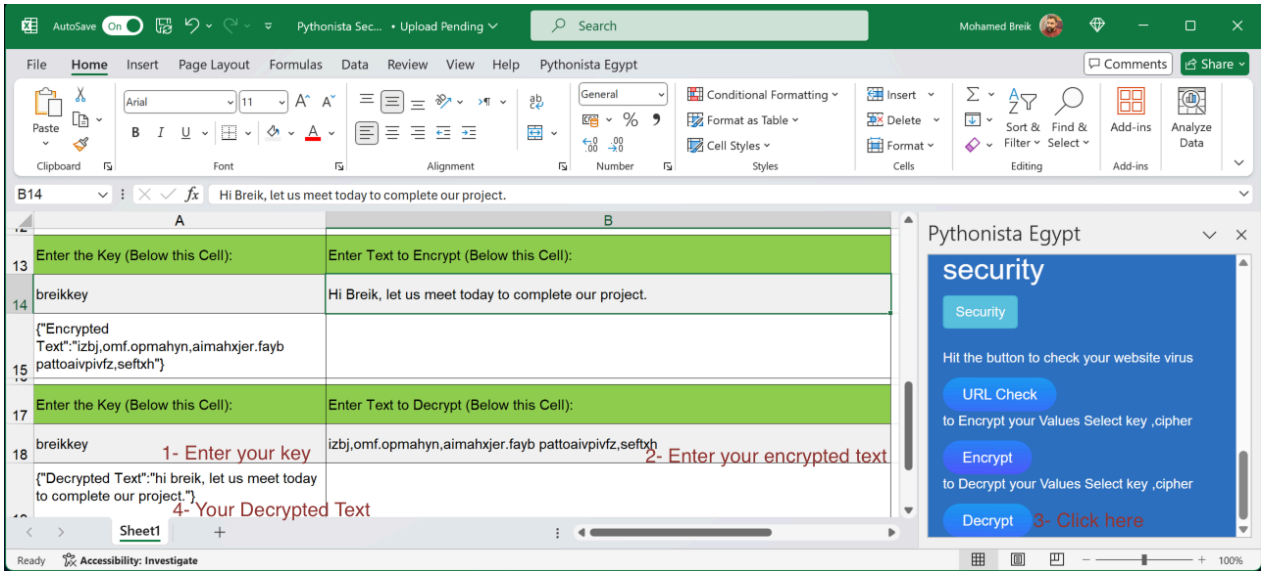

Figure 5. Decryption using Excel add-in.

• It provides a straightforward method for even non-technical users to encrypt and decrypt messages within Excel, promoting data privacy and security awareness.

5.2. Usability Enhancements

• It enhances the usability of cybersecurity tools by embedding them into Excel, an environment familiar to many users, thereby reducing the barrier to effective cybersecurity practices.

• It includes the development of user-friendly documentation and interfaces, thereby empowering users with little to no background in cybersecurity to take proactive measures to protect their data.

5.3. Technical Contributions

- It contributes to the field by demonstrating the practical integration of external security APIs within office productivity software.
- It applies and tests encryption algorithms within a new context, providing insights into their usability and performance in everyday applications.

5.4. Educational Value

- The add-in serves as an educational tool, raising awareness about the importance of cybersecurity and the risks associated with malicious URLs and unencrypted data.
- It provides a real-world application of encryption, potentially sparking user interest in the wider field of cybersecurity.

6. Future Work

For future research and development, several avenues are presented by this study:

6.1. Advanced Threat Detection

- Integrate more advanced machine learning-based models to predict and detect zero-day phishing URLs and other emerging threats.
- Explore the integration of additional security APIs to broaden the scope of threat detection.

6.2. Cross-Platform Adaptability

- Adapt the add-in for use with other spreadsheet software and platforms, increasing the tool's accessibility and utility.
- Create mobile versions of the add-in to extend protection to users on mobile devices.

Acknowledgements

We are grateful to Prof. Mervat Gheith, Prof. Essam Amin and Dr. Tarek Aly for their guidance and support. We also extend our thanks to the Pythonista Egypt group for their valuable contributions to this research.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] Manjusha, K.N.S.B.V. and Jaya Kumari, D. (2024) Detecting Phishing Links Analysis Using Machine Learning. *International Journal for Multidisciplinary Research*, **6**. https://doi.org/10.36948/ijfmr.2024.v06i03.18870

[2] Koca, M., Avci, İ. and Al-Hayani, M.A.S. (2023) Classification of Malicious Urls

Using Naive Bayes and Genetic Algorithm. *Sakarya University Journal of Computer and Information Sciences*, **6**, 80-90. https://doi.org/10.35377/saucis...1273536

[3] Barricelli, B.R., Cassano, F., Fogli, D. and Piccinno, A. (2019) End-User Development, End-User Programming and End-User Software Engineering: A Systematic Mapping Study. *Journal of Systems and Software*, **149**, 101-137. https://doi.org/10.1016/j.jss.2018.11.041

[4] Thompson, K. (1984) Reflections on Trusting Trust. *Communications of the ACM*, **27**, 761-763. https://doi.org/10.1145/358198.358210

[5] Huang, Y., Tsai, C., Lin, T., Huang, S., Lee, D.T. and Kuo, S. (2005) A Testing Framework for Web Application Security Assessment. *Computer Networks*, **48**, 739-761. https://doi.org/10.1016/j.comnet.2005.01.003

[6] Rubinstein-Salzedo, S. (2018). The Vigenère Cipher. In: Rubinstein-Salzedo, S., Ed., *Cryptography*, *Springer Undergraduate Mathematics Series*. Springer, Cham, 41-54. https://doi.org/10.1007/978-3-319-94818-8_5

[7] Choo, E., Nabeel, M., Kim, D., De Silva, R., Yu, T. and Khalil, I. (2023) A Large Scale Study and Classification of Virustotal Reports on Phishing and Malware Urls. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, **7**, 1-26. https://doi.org/10.1145/3626790

[8] Shalaginov, A., Franke, K. and Huang, X. (2016) Malware Beaconing Detection by Mining Large-Scale DNS Logs for Targeted Attack Identification. *International Journal of Computer and Information Engineering*, **10**, 743-755. https://publications.waset.org/10004242/pdf

[9] Lieberman, H., Paternò, F., Klann, M. and Wulf, V. (2006) End-User Development: An Emerging Paradigm. In: Lieberman, H., Paternò, F., Wulf, V., Eds., *End User Development*, *Human-Computer Interaction Series*, vol. 9, Springer, Dordrecht, 1-8. https://doi.org/10.1007/1-4020-5386-x_1

[10] Ko, A.J., Abraham, R., Beckwith, L., Blackwell, A., Burnett, M., Erwig, M., *et al.* (2011) The State of the Art in End-User Software Engineering. *ACM Computing Surveys*, **43**, 1-44. https://doi.org/10.1145/1922649.1922658

[11] Fischer, G., Fogli, D. and Piccinno, A. (2017) Revisiting and Broadening the Meta-Design Framework for End-User Development. In: Paternò, F. and Wulf, V., Eds., *New Perspectives in End-User Development*, Springer, Cham, 61-97. https://doi.org/10.1007/978-3-319-60291-2_4

[12] Rak, M., Ficco, M., Battista, E., Casola, V. and Mazzocca, N. (2014) Developing Secure Cloud Applications. *Scalable Computing: Practice and Experience*, **15**. https://doi.org/10.12694/scpe.v15i1.965

[13] Ghumatkar, R.S. and Date, A. (2023) Software Development Life Cycle (SDLC). *International Journal for Research in Applied Science and Engineering Technology*, **11**, 1162-1165. https://doi.org/10.22214/ijraset.2023.56554