Prospectus

Enhancing Code: Refactoring and Adding Type Hints
(Technical Report)

The Role of Passwordless Authentication in Creating a More Secure World
(STS Research Paper)

Presented to the Faculty of the School of Engineering and Applied Science
University of Virginia • Charlottesville, Virginia

In Fulfillment of the Requirements for the Degree
Bachelor of Science, School of Engineering

Matthew Beck
Fall, 2023
Department of Computer Science

Advisors

Prof. S. Travis Elliott, Department of Engineering and Society

Prof. Briana Morrison, Department of Computer Science

Introduction

My technical capstone project details real-world examples of problems encountered within an agile software development environment and how they are combatted. This topic and subsequent findings are rooted in my experience this summer at my internship. I made my team's code type-safe and refactored the code to ensure efficiency and security.

This STS thesis will focus on the sociotechnical aspects of implementing passwordless authentication systems. The social factors at play hold influence over the implementation of this technology. These social factors include benefits such as increased security and concerns such as privacy issues.

Technical Topic - Enhancing Code: Refactoring and Adding Type Hints

Abstract

When working in an agile environment, a company's repository can become dense and repetitive. Additionally, the incorrect types of variables can be used when using Python. To combat this, I refactored the code to eliminate redundant functionality sourced from different locations in the code. I added type hints to make the code safer.

Additionally, I went through my team's entire API and added type hints to the method signatures. By refactoring the code, I eliminated hundreds of lines of code and made the repository less dense. Adding type hints enabled developers to perform static type checking while enhancing autocompletion in developers' IDE. In the future, more code must be refactored, and more type hints must be added.

Introduction

Group work often leads to chaos, and certain aspects must be remembered. This is no different in the space of agile software development. Multiple people work on various facets of the project simultaneously, resulting in a code repository that begins to be filled with redundant code. Additionally, when using a dynamically typed language such as Python, the project becomes vulnerable to incorrect types used throughout the code, introducing bugs.

This past summer, I worked at a technology firm based in Boston. I quickly joined a team and was assigned to work on their testing platform. These problems were apparent throughout their code. The same functionality was written in multiple locations, all of the callers of a function performed the same actions post-call, and incorrect types were being passed in various locations.

These issues are critical to address because of the bugs that can be introduced and the overall decrease in productivity for the developers. Transforming a repository to have concise, type-safe code employs developers to work efficiently and allows them to focus on their projects. My approach to this problem was adding type hints to their testing API while refactoring their code to have related functionality in the same place.

Related Works

The dynamically typed state of Python and the problems that may arise because of it are very well documented. The issues that arise involve scoping problems and a need for early feedback (Lewis, 2021). These problems can be combated with Python's introduction of type hinting in version 3.5. According to CodersLegacy, adding type hints to your Python code has

various benefits, such as improving readability, debugging, and IDEs. There are some issues with instituting type hints, such as the time it takes to add post-production and an increase in startup time.

The concept of refactoring code has been around since the beginning of computer science, with its benefits heavily researched and documented. In Gillis' article, they detail precisely how refactoring not only makes the developers' lives more accessible but also makes the code better and less prone to bugs. Refactoring code has been shown to increase readability while reducing complexities. Refactoring the code becomes cleaner, more efficient, and maintainable. Perhaps the biggest reason to refactor code is the benefit of developers knowing where standard core functionality lies, significantly improving efficiency and allowing them to focus on their problem.

Moving Forward

To complete my technical report, I plan to detail my approach to this problem and my experience in approaching it. I plan on documenting the dynamics of refactoring code that a whole department of people is working on. I will explain the complexities of communication and the technical aspect of deciding which code to refactor and which not to refactor. Additionally, I want to detail the challenges that arise in type-hinting already written code and the benefits I found that were provided to the company.

Finally, I wish to show the results I experienced as a product of my work, what benefits were provided, and what issues came with refactoring code and adding type hints for myself and the company.

References

Benefits of Type Hinting in Python. (n.d.). CodersLegacy. Retrieved September 29, 2023, from https://coderslegacy.com/python/benefits-of-type-hinting/

Gillis, A. S. (n.d.). *What is Refactoring (Code Refactoring)?* TechTarget. Retrieved September 29, 2023, from https://www.techtarget.com/searchapparchitecture/definition/refactoring

Lewis, M. (2021, December 9). *The Struggle of Dynamically Typed Languages | by Mark Lewis | Medium*. Mark Lewis. Retrieved September 29, 2023, from https://drmarkclewis.medium.com/the-struggle-of-dynamically-typed-languages-ef91a87164a1

STS Thesis - The Role of Passwordless Authentication in Creating a More Secure World

Introduction

This thesis aims better to understand passwordless authentication's social construction and adoption. My approach to accomplish this involves applying the Social Construction of Technology (SCOT) framework to the environment where passwordless authentication is currently implemented. The popularity of passwordless authentication has been increasing, which allows the evaluation of which social aspects favor the adoption and which social factors are related to the reluctance to adopt this system. Among those who prefer passwordless authentication are those who yearn primarily for the increase in security associated with passwordless authentication, but concerns are prevalent regarding privacy and the concept of change.

Background

The most common form of web authentication used is passwords. Technopedia defines a password as a string of characters used for authentication. However, there are many issues

revolving around passwords. Many users choose weak passwords, making them vulnerable to being compromised by malicious actors. Additionally, many passwords are stolen through various techniques, often from users sharing their passwords or phishing attempts. Passwordless authentication revolves around authenticating users without the need for passwords. Many different forms of passwordless authentication exist, including single sign-on and biometrics.

Passwordless authentication is a concept that has been introduced previously. The origin dates back to the mid-1960s at the Massachusetts Institute of Technology with the creation of the Compatible Time-Sharing System (CTSS) (Cisco Systems, inc. 2020). However, such systems were less preferred over the convenience and security of passwords at the time. As time progresses, passwords become a more prominent source of security vulnerabilities. Zachary Comeau details how Microsoft conducted a study that tracked nearly 1,300 password attacks every second and a rise of 61% in phishing attacks from 2021 to 2022 (Comeau, Z. 2023). This leads to the call for better security. The question then becomes how better security can be instituted when passwordless authentication has been viewed as the answer for nearly 20 years. In 2004, during a keynote speech at the RSA Security Conference, Bill Gates claimed that people would rely less on passwords over time. He explained how his company, Microsoft, had recently shifted to a biometric keycard system (Kotadia, M. 2004).

In 2012, a technical report was produced by the University of Cambridge detailing the benefits that come with passwordless authentication. A comprehensive list of security benefits is among the benefits listed, including resilience to theft and no trusted third party in the passwordless authentication process. Additionally, this technical report makes claims of ease of use and compatibility across systems, but concerns arise from these claims (Bonneau, J., Herley, C., van Oorschot, P. C., & Stajano, F. 2012).

There are many concerns among users rooted in the adoption, privacy concerns, and overall compatibility. Many users are not willing to implement this change. According to a study by the HIPAA Journal, 55% of respondents prefer relying on memory for passwords (Alder, S. 2023). Forbes detailed how change is often met with resistance by users and passwordless authentication is uncharted territory. Additionally, Forbes recognized the privacy concerns held by some stakeholders, saying that the lack of clarity on where the data is stored is problematic for users (Singh, A. 2019).

Approach

The Social Construction of Technology (SCOT) framework responded to technological determinism. The central idea behind the SCOT framework is that social factors dictate how technology is created and implemented. The main concepts of SCOT needed for the analysis of passwordless authentication include relevant social groups, design flexibility, and closure.

The relevant social groups are the stakeholders who make decisions about the technology and anyone who is affected by the technology. In the case of passwordless authentication, the largest relevant social group is that of the users. The users are going to be the main group that will be in favor of or reluctant to implement a new system. In addition to the users, the developers must also be considered. The social group of developers typically favors passwordless authentication due to its increased security. Lastly, the businesses and structures currently holding password authentication must be considered. It must be regarded as how the environments will be affected when there is a change in the authentication methodology.

Interpretive flexibility primarily deals with how a problem can be defined in different ways. In the case of passwordless authentication, the intepretive flexibility lies within the belief

of the problem's existence. With passwordless authentication's primary objective of addressing the security issues that lie within authentication with passwords, it is imperative that all of the relevant social groups identify this problem. However, this is not the case when it comes to authentication without passwords. Many users fail to see issues regarding passwords and therefore do not see the need for the introduction of passwordless authentication. On the other hand, many developers along with some users recognize the issues with passwords and see alternatives to passwords as the solution.

Closure and stabilization deals directly with the closing of this interpretive flexibility gap. In this case, closure and stabilization can only be achieved if there is a general consensus that recognizes the issues regarding passwords. The problem may then be stabilized through either introducing passwordless authentication or through future technological advancements that will make passwords more secure.

Moving Forward

In my background section, I briefly outlined the positive and negative elements of password authentication. In my future work, I aim to draw a more extensive analysis of and clearly define the social role of these elements. I desire to explore how implementing passwordless authentication will positively affect workplaces and existing systems and the tremendous social impact generated by the implementations that will ultimately shape the technology.

To proceed, I plan to conduct more research, gather materials specific to these different elements, and learn how they have shaped passwordless authentication and will shape passwordless authentication in the future. Additionally, I would like to compile a table

explaining the advantages and disadvantages of passwords and the different forms of passwordless authentication.

Conclusion

With increasing concerns around password authentication security, the popularity of passwordless authentication has increased. Using the Social Construction of Technology framework, I examine the positive and negative elements of passwordless authentication and how these elements create a social environment that shapes how passwordless authentication is and will be implemented. In my future work, I will expand upon both the positive and negative elements and provide analysis through the SCOT framework of how it shapes and impacts this technology.

References

Alder, S. (2023, May 4). *Passwordless Authentication Adoption Increases but Poor Password Practices Persist*. The HIPAA Journal. Retrieved October 4, 2023, from https://www.hipaajournal.com/passwordless-authentication-poor-password-practices/

Bonneau, J., Herley, C., van Oorschot, P. C., & Stajano, F. (2012, March). *The quest to replace passwords: a framework for comparative evaluation of Web authentication schemes*. The University of Cambridge. UCAM-CL-TR-817

Cisco Systems, inc. (2020). *Passwordless* [The Future of Authentication] [ebook]. Enterprise Talk. Retrieved 10 4, 2023, from https://resources.enterprisetalk.com/ebook/Duo-224-EN-7.pdf

Comeau, Z. (2023, May 5). *Passwordless Tools Are On The Rise, But Adoption Will Take Time*. My TechDecisions. Retrieved December 4, 2023, from https://mytechdecisions.com/network-security/passwordless-tools-on-the-rise-adoption-will-take-time/

Kan, M. (2023, April 26). *Fragmentation, Lack of Adoption Impede Uptake of PasswordLess Logins*. PCMag. Retrieved October 4, 2023, from

https://www.pcmag.com/news/fragmentation-lack-of-adoption-impede-uptake-of-passwordless-logins

Kotadia, M. (2004, February 25). *Gates predicts death of the password | ZDNET*. ZDNet. Retrieved October 4, 2023, from https://www.zdnet.com/article/gates-predicts-death-of-the-password/

Rouse, M., & Keary, T. (2023, September 6). *What is a Password? - Definition from Techopedia*. Techopedia. Retrieved October 5, 2023, from https://www.techopedia.com/definition/4042/password

Singh, A. (2019, March 9). *';;'*. Why is Passwordless Authentication Met with Reluctance? Retrieved October 4, 2023, from https://www.forbes.com/sites/forbestechcouncil/2021/04/20/why-is-passwordless-authentication-met-with-reluctance/?sh=d5f787466d06