Team 5

OSS Project Overview

For our project we have chosen the Open-Source Software BitWarden for its use as a password manager, secrets managers, and passkey innovator. The software can be used by a business to manage their employee passwords and help protect the customer or client data stored on company servers. We will be utilizing both the client based BitWarden, which includes the desktop, web, and browser extension but not the mobile application, as well as the BitWarden server infrastructure and backend. The software offers a password manager that can be used by all the employees of our hypothetical operations environment as well as an account manager which allows for the automation of user accounts from a centralized admin panel. This password manager is cross-platform and allows for access from an unlimited number of devices. The software also offers a secrets manager, which our hypothetical operations environment access to and end-to-end encryption to secure their server infrastructure and code. This secrets manager gives centralized storage and has clear audit trails of secret access operations.

Hypothetical Operations Environment

Our hypothetical operations environment consists of a small web development team with a webpage, e-commerce, and backend server components. This web development team needs access to security solutions that will protect the DMZ that holds their public facing webpage, as well as their backend development resources and servers. This team might also expand in the future, assuming they do not currently possess off-site personal, and the end-to-end encryption the secrets manager offers would allow them to protect their data in transit. The centralized password manager and admin control panel allows an admin to manage their entire staff as they expand from a single console that is protected locally.

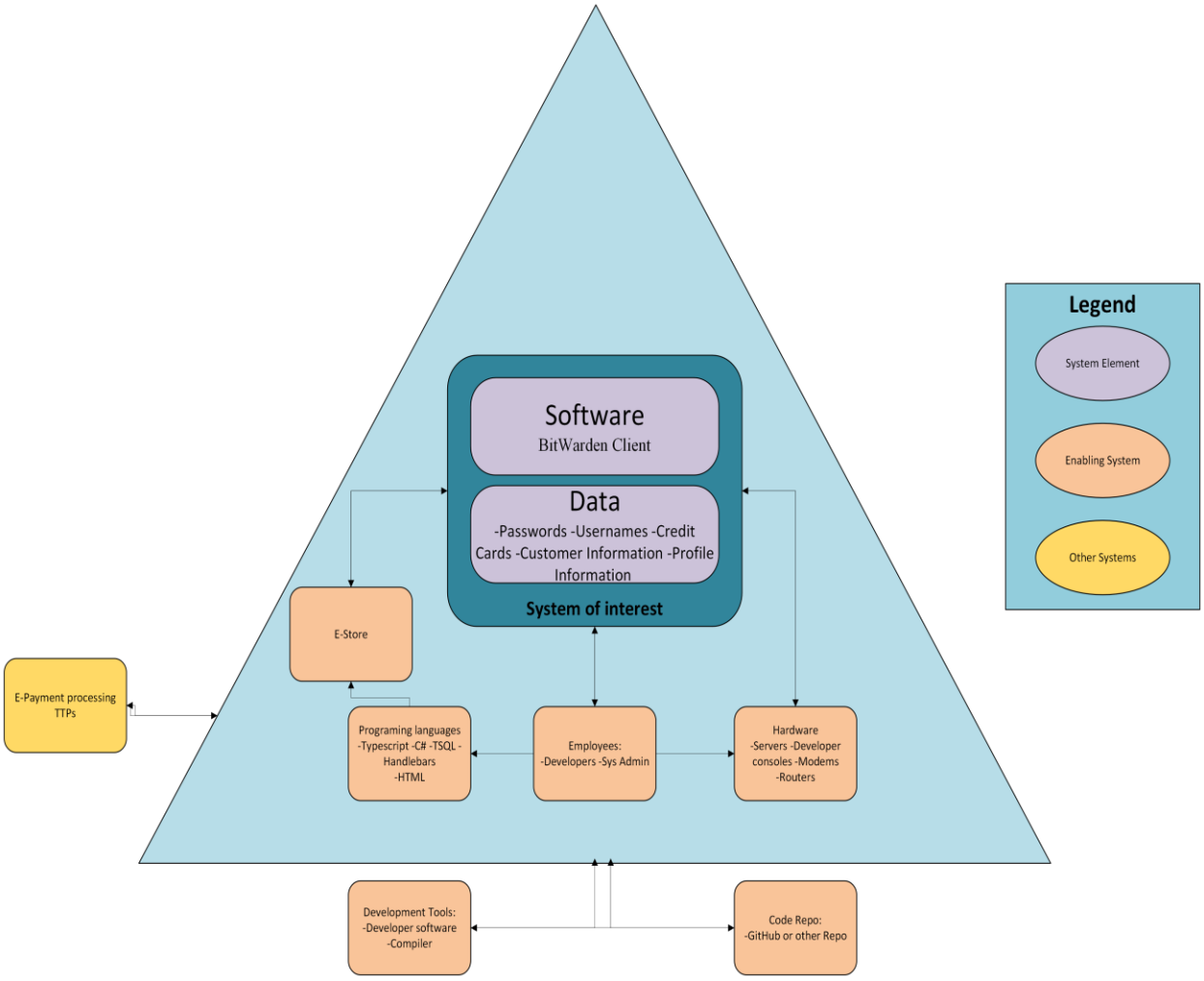
Our Motivation

Choosing Bitwarden as a research project is motivated by several factors, especially in areas of cybersecurity, software development, open-source ecosystems, and user experience. Here are some key motivations:

With the increasing number of data breaches, identity theft, and password-related cybercrimes, researching password managers like Bitwarden addresses a critical area of cybersecurity. Bitwarden is fully open source, which provides access to the complete codebase for analysis, testing, and modifications. It allows researchers to study its architecture, encryption methods, or even contribute to the project. Bitwarden works on various platforms, making it a good candidate

for studying cross-platform development and usability. Bitwarden can be compared to other popular password managers like LastPass, 1Password, and Dashlane, focusing on security, cost, usability, or features. Bitwarden's community-driven development allows researchers to study how collaboration and contributions in open-source projects influence product security and innovation. Bitwarden offers free and paid versions, creating an opportunity to study its business model and the economic viability of open-source password managers. Researchers can perform vulnerability assessments, penetration testing, or cryptographic reviews on Bitwarden to evaluate its real-world security.

SED



- Perceived Threats

Bitwarden is a popular open-source password manager, but like any software, it may have perceived or potential threats. Here are some of the common concerns:

Bitwarden stores encrypted vaults on their cloud servers, raising concerns about the risk of cloud server breaches or attacks. Users may fear that the encryption algorithms or key management could be weaker than expected, leading to possible decryption by malicious actors. Integrations with other services or plugins might introduce vulnerabilities, especially if third-party services are compromised or misconfigured. Since Bitwarden's code is open source, some worry that this transparency makes it easier for attackers to find and exploit vulnerabilities. Users may set weak master passwords, making their vault more vulnerable to brute force or phishing attacks. A user might accidentally give their master password to a fake Bitwarden login page or fall victim to social engineering. If Bitwarden's servers are compromised, attackers could try to access encrypted vaults. Since Bitwarden syncs passwords across devices, some users worry about interception during syncing. Some users might question whether Bitwarden fully adheres to the "zero-knowledge" model and if any backdoors could exist.

- Security Features

Bitwarden is known for its robust security features designed to protect users' sensitive information, particularly passwords. Here are the key security features: End-to-End Encryption, Zero-Knowledge Architecture, Two-Factor Authentication (2FA), Master Password Requirements, Secure Password Sharing, Password Generator, Self-Hosting Option, Vault Health Reports, Security Audits, Cross-Platform Sync with Secure Transmission, Emergency Access, Encrypted File Attachments, Biometric Login, Data Breach Alerts, Audit Trail and Access Control (for Organizations), Single Sign-On (SSO) Integration, API and Command-Line Interface (CLI) Security, and Session Timeout and Auto-Logout.

These security features make Bitwarden a strong choice for managing passwords and sensitive data, with a heavy focus on encryption, user control, and flexibility. The security features of Bitwarden are designed with durability and long-term security in mind, using industry best practices and highly regarded cryptographic standards.

Security History of Bitwarden

Bitwarden uses a zero-knowledge architecture which means the service doesn’t store or access the master password or any decrypted data. It uses end-to-end encryption to make sure that sensitive data is encrypted locally on the user's device before it is transmitted onto Bitwardens servers. AES-256 Encryption, PBKDF2 for key derivation, and HMAC-SHA256 for hashing are key components of Bitwardens cryptographic framework. In 2018 Bitwarden underwent its first

security audit which was conducted by Cure53. They are a well-known cybersecurity firm. This audit was not open to the public which led to very few issues, and they were quickly patched. This helped increase confidence in Bitwarden security architecture, reinforcing their zero-tolerance policy and their secure encryption policy. As Bitwarden grew they introduced two-factor authentication with support for Yubikey, Duo, Universal Second Factor. Also to make sure that the scalability was good, Bitwarden started doing regular penetration testing, and audits by independent security firms. From 2020-2022, there were large-scale data breaches and incident response reports. Bitwarden was able to maintain a strong record of zero known breaches. They also use a disclosure program where researchers can report vulnerabilities, which leads to them being fixed. In the wake of these incidents Bitwarden was more recognized because of well their security with their open-source transparency, and public audits.

Licensing Information

Bitwardens source code is released under the GNU General Public Liscense v3.0 which allows users to use it freely, modify it, and distribute the software. The full source code must be available if it is redistributed. This helps with transparency and trust. If people are going to make significant contributions, they may be required to sign a Contributor License Agreement. This makes sure that the owners have copyright over their contributions.

Requirements for Software Security Engineering

For a vigorous software security operation, the focus should be on protecting user data, ensuring secure coding practices, and maintaining system integrity. The goal is to implement encryption for vault data both at rest and in transit while ensuring data is encrypted locally before it is sent to the server. We will use secure, user-controlled encryption keys derived from a strong master password using a key derivation function like PBKDF2 with a high iteration count. In the “zero knowledge” phase the goal is to let the server handle encrypted data, with decryption happening exclusively on the user’s device without transmitting any form of the master password. The ability to implement 2FA methods like TOTP, email-based 2FA, hardware-based tokens (YubiKey, FIDO U2F), enforce strong password policies, allow biometric authentication (e.g., fingerprint or Face ID) and perform regular third-party security audits will enhance the chance of the system upholding. Following best practices in secure coding (e.g., OWASP top 10) to prevent common vulnerabilities like SQL injection, cross-site scripting (XSS), and buffer overflows is essential. In the event of a security incident, notify users immediately and provide guidance on how to mitigate risk. Offer guidelines and best practices for securely deploying and maintaining a self-hosted instance. Ensure Bitwarden complies with applicable privacy and security regulations, such as GDPR (General Data Protection Regulation), HIPAA (Health

Insurance Portability and Accountability Act), and SOC 2. Have an incident response plan in place for detecting, responding to, and mitigating security breaches or vulnerabilities. Ensure encrypted backups of vaults are available and implement procedures for secure recovery in case of data loss, without exposing vault contents. By adhering to these requirements, Bitwarden can maintain a secure environment that protects users' sensitive data while remaining transparent, responsive, and resilient against emerging security threats.

GitHub Link

https://github.com/PatrickBN/CYBR8420_Team5

Reflections

Developing a system like Bitwarden involves more than just building a secure password manager. It requires continuous reflection on the balance between security and usability, maintaining user trust through transparency, adapting to future threats, and creating a system that scales for both individuals and businesses. Here are some reflections when developing such a system:

1. Balancing Security and Usability - One of the biggest challenges in developing Bitwarden is striking the right balance between security and ease of use. A password manager must provide airtight security yet be simple and intuitive enough for users to adopt without resistance. Keeping users engaged with helpful guides and warnings, without being intrusive, is critical for long-term adoption.

2. Transparency and Trust through Open-Source Development - Bitwarden's open-source nature is a cornerstone of trust and transparency. It invites scrutiny from the global security community, which in turn can identify potential vulnerabilities and suggest improvements. However, open-source development also means there's a responsibility to maintain high coding standards, thorough documentation, and clear communication with the community. As a developer, there's a need to reflect on *how to handle security disclosures*, ensuring that vulnerabilities are responsibly reported and patched, while maintaining public confidence.

3. Adapting to Emerging Threats - Developing a system like Bitwarden involves reflecting on *how the system will evolve to stay secure* against emerging threats such as quantum computing, more sophisticated phishing attacks, or new forms of malware. Regular engagement with the latest security research is essential to anticipate vulnerabilities and introduce countermeasures before they can be exploited.

4. User Empowerment and Education - Security tools are only effective if users understand and use them properly. Developing Bitwarden involves reflecting on how to empower users to take control of their security. This means developers need to keep asking, “How can we guide users to make the best security decisions?”

5. Scaling for Diverse Use Cases - Reflecting on the system’s scalability is crucial. For individuals, simplicity and quick access to vaults are priorities. For organizations, more advanced features like team password sharing, role-based access control, and secure management of employee credentials are necessary. The question to ask is “How can we ensure that Bitwarden meets the needs of users at all levels without overwhelming the software’s core simplicity?”

6. Ensuring Data Privacy in a Global Context - As Bitwarden users come from around the globe, there are reflections on how to manage data privacy across multiple jurisdictions, each with its own set of regulations. Developers need to reflect on how data is stored, managed, and encrypted to ensure compliance with international privacy standards. These decisions are not only technical but ethical as well.

7. Performance and Reliability under Security Constraints - Developing Bitwarden requires reflections on how to maintain fast, responsive performance while ensuring rigorous encryption and decryption processes. Performance optimizations need to be made in a way that does not compromise security. The question to ask is “How do we balance security with user expectations of speed and reliability?”

8. Long-Term Maintenance and Community Involvement - Developers must reflect on how to foster a supportive, active community that contributes to the project in a meaningful way.