ynt/global//global/global/global 0foks 00foks 692D399B70485E831ED596F80C068A85 0foksynt/global//global/global/global

Stash House

Franklin Diaz

September 25th, 2025

University of Colorado **Denver**

Summary

In 2023 started to focus on issues arising from a "dirty" local development environment, and precisely what to do about it. This paper is an evaluation of using freely available Open Source tools and a few easy to remember patterns we can work in a much more efficient and secure manner.

- First we look at how to set up a workable solution for storing credentials.
- Next we do some local search and cleanup to get our systems in order.
- Finally, we will see how we can still easily use these credentials though they remain encrypted at rest and in transit to our "vault" from now on.

Contents

Chapter 1

Introduction

While working as a developer, the need frequently arises to work in a "sandbox" type environment, where security controls are often skipped in favor of speed and efficiency. During the course of the software development life cycle, the need arises to give access to sensitive credentials including databases, cloud accounts, certificates, and many more.

Chapter 2

Safety Considerations for Data at Rest

The main premise of this paper is to improve the hygiene of local credential storage. With this goal in mind, we approach the question of how to meet our storage goal.

2.1 Local Back End Storage

While the tool chain set forth in this paper will use the GNU Privacy Guard to encrypt user secrets for safe storage, an additional security safeguard can be implemented by simply storing your data in a "local" fashion. That is to say, the encrypted data stays within the Enterprise Network or Administrative Domain, with the goal of reducing it's accessibility those who are unauthorized.

2.2 Distributed Back End Storage