Application Security (BTI2018) Vula Project Proposal: Enhancing the CLI for System Administrators

Vaney Quentin Guillaume, Noah Ferrari, Joel Meier

March 20, 2025

1 Introduction

Vula provides a command-line interface (CLI) for interacting with its functionality, but its current implementation lacks the flexibility and usability needed for system administrators. The goal of this project is to improve the Vula CLI to make it more scriptable, accessible, and efficient for system administration tasks. We aim to redesign the CLI to support automation in scripts, integration with configuration management tools such as Ansible, and to ensure accessibility, particularly for colorblind users. Accessibility is a crucial aspect of security, as excluding certain users from efficiently managing a system introduces risks and inefficiencies.

2 User Story: The System Administrator's Perspective

Frank Hank, a system administrator in a medium-sized organization, needs to deploy and maintain multiple Vula instances across different environments. He wants to:

- Automate Vula deployments using shell scripts and Ansible playbooks.
- Ensure that CLI output is structured, machine-readable (e.g., JSON or YAML output), and follows UNIX conventions.
- Have clear and unambiguous CLI output that works well for users with different types of color vision deficiencies by offering multiple configurable color schemes.
- Utilize autocomplete functionality for commands and arguments to improve usability and efficiency.

With the current CLI, he faces challenges due to inconsistent output, lack of machine-readable formats, poor accessibility for users with color vision deficiencies, and a lack of autocomplete functionality. By improving the CLI, Frank Hank and other system administrators can integrate Vula into automated workflows with greater reliability and security.

3 Objectives

1. Design and implement a more script-friendly Vula CLI.
2. Ensure output formatting supports structured data (e.g., JSON/YAML) for better automation.
3. Improve accessibility by ensuring CLI messages offer multiple color schemes optimized for different types of color blindness.
4. Provide detailed documentation and usage examples for automation scenarios.
5. Implement autocomplete functionality to enhance CLI usability.

The minimum deliverables for this project are objectives 1, 2, and 5, ensuring that system administrators can use the CLI effectively in automation scripts.

4 Security and Accessibility Considerations

Accessibility is an important aspect of security. If a CLI is difficult to use for colorblind users, critical errors may be missed, increasing the likelihood of misconfigurations and security incidents. Our approach to accessibility includes:

- Avoiding reliance on color alone by offering multiple predefined color schemes optimized for different types of color blindness, including deuteranopia, protanopia, and tritanopia.
- Providing visual distinctions through symbols, text labels, and patterns alongside color.
- Ensuring clear, structured error messages that aid in troubleshooting.
- Supporting configurable accessibility options to adapt the CLI to various user needs.

Additionally, improving scriptability and autocomplete functionality enhances security by reducing manual errors in deployments and configurations, allowing for better consistency and auditability.