Simplifying Source Code Signing
The Redesign of GPGit

Exposé term paper

NicoHood

April 23, 2017

Abstract

Digital signatures enable easy verification of source code authenticity and integrity. There is a widespread belief that using digital signature software is difficult and time-consuming. In order to make digital signing ubiquitous, this term paper describes a strategy and the development of GPGit from its first to its second version for accomplishing this goal.

Contents

List of Figures

1 Introduction

1.1 Motivation

As we all know, today more than ever before, it is crucial to be able to trust our computing environments. Malicious software (henceforth malware) continues to be a major security threat and can have a *major economic impact*[1] on home and enterprise users. They must be able to detect if a software application obtained from another computer system was tampered.

One way to ensure that the downloaded software contains no malware is to transfer it over a secure connection and to verify its digital signature. *GPG*[2] offers a quick and easy way to *sign and verify source code releases*[3] using *digital signatures.*[4]

1.2 Problem Description

Most software developers do not provide digital signatures for their software releases. There is a widespread belief that using digital signature software is *difficult and time-consuming.*[5]

Since most developers lack knowledge or are misinformed about cryptography and security matters they are unaware that digital signature software exists or they avoid using it.

A lot of *FLOSS*[6] was and is developed by hobbyist developers in their free time. Since most develop the software for personal use and enjoyment, they are less willing to adopt security measures. In contrast to that, companies often have time requirements that conflict with other goals.

1.3 Aims and Objectives

This project aims to educate developers about the importance of digital signature software and to increase the number of its users. This will in turn reduce the risk of malware spreading.

[1] https://us.norton.com/cyber-security-insights-2016
[2] https://www.gnupg.org/
[3] https://www.gnupg.org/gph/en/manual/x135.html
[4] https://www.gnupg.org/gph/en/manual/x215.html
[5] https://moxie.org/blog/gpg-and-me/
[6] https://www.gnu.org/philosophy/floss-and-foss.en.html

GPGit (which is described below) was created to simplify the digital signing task for both beginners and advanced users. In order to make digital signing ubiquitous, this term paper describes a strategy and the development of GPGit from its first to its second version for accomplishing this goal.

2 The GPGit Project

2.1 Design Principles

2.2 Potential of Improvements

3 The Redesign of GPGit

3.1 User Interface

3.2 Parameters & Configuration

3.3 Source Code Style

3.4 User Documentation

3.5 Implementation Details

4 Conclusion

5 Glossary