Avoiding the Top 10 Security Flaws

Introduction

Most software that has been built and released typically comes with a set of defects — implementation bugs and design flaws. To date, there has been a larger focus on finding implementation bugs rather than on identifying flaws.

In 2014, the IEEE Computer Society, the leading association for computing professionals, launched a cybersecurity initiative with an aim to make an impact in the field of cybersecurity. The first step for the initiative was to launch the IEEE Center for Secure Design. The Center intends to shift some of the focus in security from finding bugs to identifying common design flaws in the hope that software architects can learn from others' mistakes. To achieve this goal, the Center brought people together from different organizations at a workshop in early 2014.

At the workshop, participants discussed the types of flaws they either identified in their own internal design reviews, or that were available from external data. They arrived at a list they felt were the top security design flaws. Many of the flaws that made the list have been well known for decades, but continue to persist. In this document is the result of that discussion — and how to avoid the top ten security flaws.

Center for Secure Design participants

- Iván Arce, Sadosky Foundation
- Neil Daswani, Twitter
- Jim DelGrosso, Cigital

- Danny Dhillon, RSA
- Christoph Kern, Google
- Tadayoshi Kohno, University of Washington
- Carl Landwehr, George Washington University
- Gary McGraw, Cigital
- Brook Schoenfield, Intel/McAfee
- Margo Seltzer, Harvard
- Diomidis Spinellis Athens University of Economics and Business
- Izar Tarandach, EMC
- Jacob West, HP

Mission Statement

The IEEE Computer Society's Center for Secure Design (CSD) will gather software security expertise from industry, academia, and government. The CSD provides guidance on:

- Recognizing software system designs that are likely vulnerable to compromise.
- Designing and building software systems with strong, identifiable security properties.

The CSD is part of the IEEE Computer Society's larger cybersecurity initiative, launched in 2014.

Preamble

The goal of a secure design is to enable a system that supports and enforces the necessary authentication, authorization, confidentiality, data integrity, accountability, availability, and non-repudiation requirements, even when the system is under attack.

While a system may always have **implementation defects or "bugs,"** we have found that the security of many systems is breached due to **design flaws or "flaws"**. We believe that if organizations design secure systems, which avoid such flaws, they can significantly reduce the number and impact of security breaches.

While **bugs** and **flaws** are both different types of **defects**, we believe there has been quite a bit more focus on common bug types than there has been on secure design and the avoidance of flaws. Before we discuss our contribution in this document, we briefly discuss the differences between bugs and flaws.

Both bugs and flaws are types of **defects**. A defect may lie dormant in software for years only to surface in a fielded system with major consequences. A **bug** is an implementation-level software problem. Bugs may exist in code but never be executed. A **flaw**, by contrast, is a problem at a deeper level. Flaws are often much more subtle than simply an off-by-one error in an array reference or use of an incorrect system call. A flaw might be instantiated in software code, but it is the result of a mistake or oversight at the design level. For example, a number of classic flaws exist in error-handling and recovery systems that fail in an insecure or inefficient fashion.

In this document, a group of software security professionals have contributed both real-world data and expertise to identify some of the most significant ***design flaws*** that have led to security breaches over the past several years. The list of issues presented here is focused entirely on the most widely and frequently occurring design flaws as compiled by data provided by the member organizations of the IEEE Center for Secure Design (CSD).