DIVERSITY

John Knight
University of Virginia
Charlottesville, Virginia, USA

Abstract

Diversity works well in nature where it is the basis of natural selection, a phenomenon that helps biological populations survive as they are challenged by hazards in their environments. Diversity also has a long history in engineering where it is used to counter the effects of design faults. Engineered systems are subject to failure, and significant losses can result from the failure of safety- and security-critical applications. A system that includes identical replicates of one or more components can survive *degradation* faults, i.e., faults that arise during operation as components age. But identical replicates do not help a system to survive *design* faults, i.e., faults that are the result of defects in the basic design. Identical replicates will contain the same defect and so will fail together on the same inputs. All software faults are design faults, because software faults are not the result of software “wearing out” over time. Defects that arise in requirements, specification and coding of software are all design faults.

A variety of different types of diversity have been developed to deal with design faults. Design diversity couples together systems with identical functionality but with different designs. The different systems are referred to as *versions*, and the versions are executed in parallel with the results subject to a vote. If erroneous outputs are produced because of design defects in some of the versions, the correct outputs will be produced provided the erroneous outputs are in a minority. There is no guarantee that the different designs will not contain the same faults, and so voting could select an erroneous output. Data diversity couples together identical *copies* of a given system but executes them in parallel with transformed data. The inverse transformation is applied to the outputs. Artificial

diversity applies an algorithmic transformation, such as relocating the address space by a random amount, to a system thereby producing *variants* that differ in a systematic way. Artificial diversity is an effective method of avoiding the “software monoculture”.

All forms of diversity have been applied successfully in the field of cyber security. Artificial diversity is especially important because: (a) when applied carefully it transforms information useful to an attacker, such as the fixed and known locations of variables, into a high-entropy search problem, (b) it incurs little to no execution-time overhead, and (c) it is applied mechanically – no development effort is required. Artificial diversity has been shown to provide strong security protection to systems that contain certain classes of vulnerability whether the problem vulnerabilities are known or unknown.

A unique characteristic of artificial diversity is that artificially diverse variants can be constructed and combined into an operational system with a property known as secretless security. For certain classes of vulnerability, such a system is provably protected against attacks and no secrets need to be kept.

CCS Concepts

• Security and privacy → Systems security; Software and application security; Software security engineering

Keywords

Design diversity; data diversity; artificial diversity; secretless security; software security

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the Owner/Author(s). Copyright is held by the owner/author(s).

CODASPY'16, March 9–11, 2016, New Orleans, LA, USA.

ACM ISBN: 978-1-4503-3935-3/16/03.

DOI: <http://dx.doi.org/10.1145/2857705.2857728>