EffectiveSan: Dynamically Typed C/C++

Gregory J. DUCK

Research Fellow

TSUNAMi, National University of Singapore

ABSTRACT

Low-level programming languages with weak/static type systems, such as C and C++, are vulnerable to errors relating to the misuse of memory at runtime, such as (sub-) object bounds overflows, (re)use-after-free, and type confusion. Such errors account for many security and other undefined behavior bugs for programs written in these languages.

In this talk, we introduce the notion of dynamically typed C/C++, which aims to detect such errors by dynamically checking the "effective type" of each object before use at runtime. We also present an implementation of dynamically typed C/C++ in the form of the Effective Type Sanitizer (EffectiveSan).

EffectiveSan enforces type and memory safety using a combination of low-fat pointers, type meta data and type/bounds check instrumentation---and is able to detect classes of errors missed by related tools. We also show that EffectiveSan achieves very high compatibility and is able to analyze large complex software such as FireFox.

SPEAKER BIOGRAPHY

Gregory J. Duck received his BSc (Mathematics) and BEng (Software) from the University of Melbourne in 2002, and his Phd (Computer Science) in 2006, also from the University of Melbourne. From 2005-2009 he was a researcher for National ICT Australia (NICTA) working for the G12 project. For 2009-2010 he worked for Constraint Technologies in Melbourne. From 2011 onwards he works at the National University of Singapore.