RESEARCH REVIEW 2019

Recovering Meaningful Variable Names in Decompiled Code

Introduction

Conventional wisdom tells us that when a compiler transforms a program from source code to an executable file, some information is lost and cannot be recovered. For example, variable names are not included in a compiled executable, and we often assume they are lost. Although state-of-the-art decompilers can recover the presence of variables, they make no attempt to recover their original names. Instead, they name the variables “v1,” “v2,” and so on. Renaming the variables is unfortunate because, as several studies have shown, programmers carefully select variable names to make the program easier to understand.

In this project, we showed that the conventional wisdom that variable names cannot be recovered is wrong. Specifically, we showed that variable names can largely be predicted based on the context of code in which they are used and accessed. We trained a neural network to predict variable names on a large corpus of C source code that we collected from GitHub.

Corpus

To generate our corpus, we scraped GitHub for projects written in C. We then automatically built 164,632 binaries from these project and extracted 1,259,935 functions. For each function, we generated a corpus entry that consisted of the original source code with placeholder variables, as shown in the code figure to the right.

Each corpus entry also included a mapping from a placeholder variable to the original identifier in the source code and the decompiler’s identifier.

We can make **exact** predictions for **74.3%** of variable names in decompiled executable code by training a neural network on a large corpus of C source code from GitHub.

```
void *file_mmap(int v1|fd|fd, int v2|size|size)
{
void *ptr|ret|buf;
ptr|ret|buf = mmap (0, v2|size|size, 1, 2, v1|fd|fd, 0);
if (ptr|ret|buf == (void *) -1)
{ perror ("mmap"); exit(1); }

return ptr|ret|buf;
}
```

Key

🟥 Decompiled 🟩 Original 🟦 Recovered

Results

Experiment	Accuracy
Overall	74.3
Function in Training	85.5
Function not in Training	35.3

When evaluating a solution based on machine learning such as ours, it is important to consider the construction of the training and testing sets. Each binary was randomly assigned to either the training or testing set. As in real reverse-engineering scenarios, library functions may be present in multiple binaries and may therefore be present in both the training and testing sets. To better understand the effect of the presence of library functions on our system, we partitioned our testing set into the set of functions that were also in the training set and those that were not in the training set. As shown in the table above, DIRE achieves 85.5% accuracy on functions it has been trained on, compared to 74.3% overall. For functions that it has not encountered in training, it yields 35.3% accuracy.

For more information, see: Jeremy Lacomis, Pengcheng Yin, Edward J. Schwartz, Miltiadis Allamanis, Claire Le Goues, Graham Neubig, and Bogdan Vasilescu. DIRE: A Neural Approach to Decompiled Identifier Renaming, Procedings of the 2019 IEEE/ACM International Conference on Automated Software Engineering.

Carnegie Mellon University
Software Engineering Institute

Bogdan Vasilescu | vasilescu@cmu.edu
Edward J. Schwartz | eschwartz@cert.org

This material is based upon work funded and supported by the Department of Defense under Contract No. FA8702-15-D-0002 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.
The view, opinions, and/or findings contained in this material are those of the author(s) and should not be construed as an official Government position, policy, or decision, unless designated by other documentation.

NO WARRANTY. THIS CARNEGIE MELLON UNIVERSITY AND SOFTWARE ENGINEERING INSTITUTE MATERIAL IS FURNISHED ON AN "AS-IS" BASIS. CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT.

DM19-1026