Final Project Progress Report

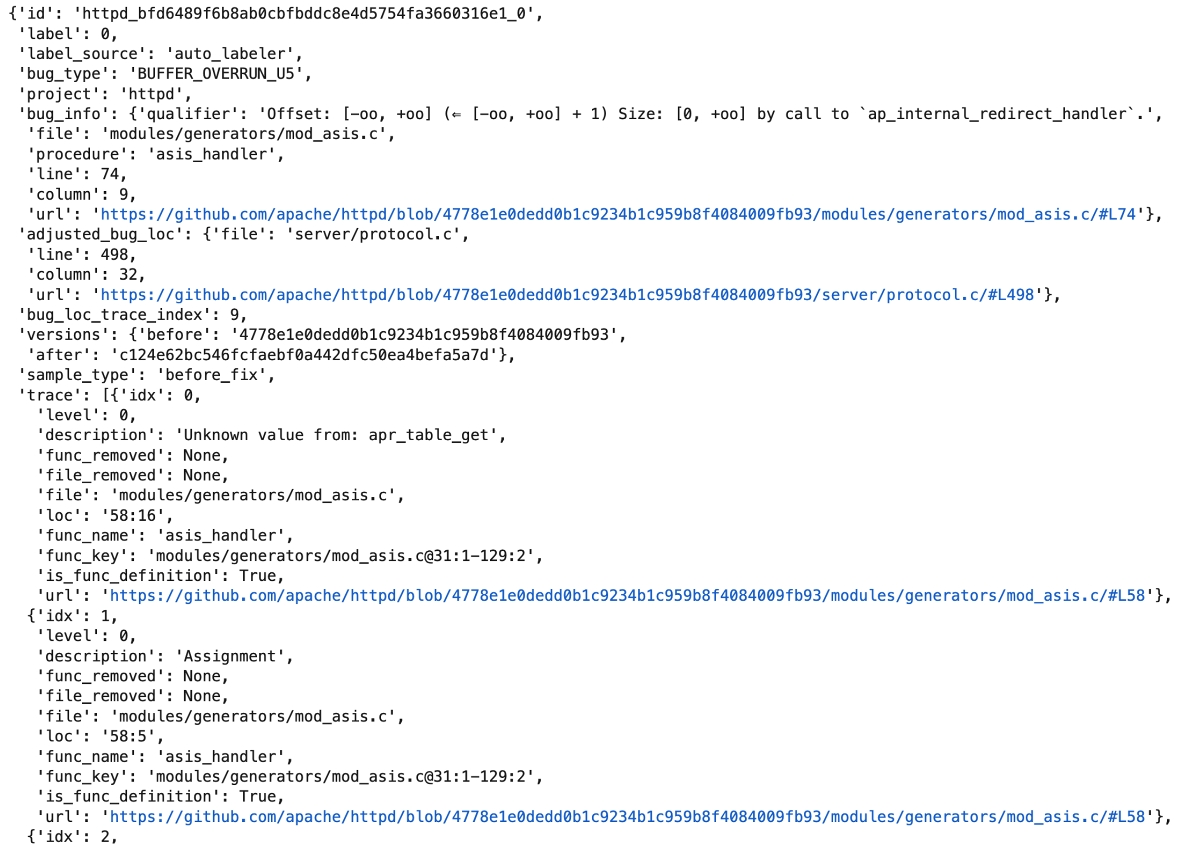
Shuo Liu, sl4921@columbia.edu

Apr. 12, 2022

1 Overview

With the increasing number of open source projects in recent years, software vulnerabilities, i.e. security flaws in computer systems, have increased and hindered the development of the industry. There are usually two types of tools for automatic bug detection, static and dynamic detection tools. Static analysis tools are often used for vulnerability detection because they can deal with large projects time efficiently. And unlike dynamic bug detection, they do not consume large amounts of computing resources to run the code. However, static code inspection tools are limited in real-world applications because of insufficient code snippets, limited data, and synthesized source code. D2A trains models for vulnerability identification by analyzing datasets constructed from differences in patched code from multiple open source projects. If a bug is not detected after a certain commit, then it is likely to be fixed by this commit. This project attempts to build a natural language deep learning model based on the D2A dataset and try to predict whether there are bugs by analyzing the functional fragments of the code [2]. The target audience are software security research community, some product divisions of large companies, and even natural language processing community, as researchers and software developer can save a lot of efforts for debugging, and it also do a favor by saving computational resource when deploying large project.

2 Research Questions

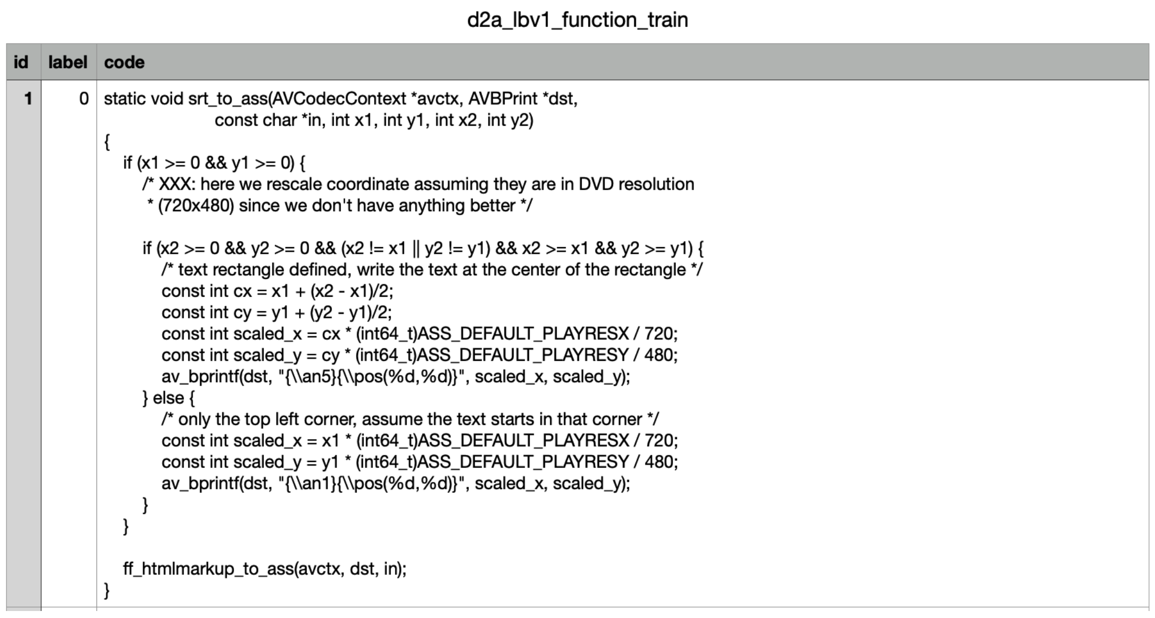


```
{'id': 'httpd_bfd6489f6b8ab0cbfbddc8e4d5754fa3660316e1_0',
 'label': 0,
 'label_source': 'auto_labeler',
 'bug_type': 'BUFFER_OVERRUN_U5',
 'project': 'httpd',
 'bug_info': {'qualifier': 'Offset: [-oo, +oo] (⇐ [-oo, +oo] + 1) Size: [0, +oo] by call to `ap_internal_redirect_handler`.',
  'file': 'modules/generators/mod_asis.c',
  'procedure': 'asis_handler',
  'line': 74,
  'column': 9,
  'url': 'https://github.com/apache/httpd/blob/4778e1e0dedd0b1c9234b1c959b8f4084009fb93/modules/generators/mod_asis.c/#L74'},
 'adjusted_bug_loc': {'file': 'server/protocol.c',
  'line': 498,
  'column': 32,
  'url': 'https://github.com/apache/httpd/blob/4778e1e0dedd0b1c9234b1c959b8f4084009fb93/server/protocol.c/#L498'},
 'bug_loc_trace_index': 9,
 'versions': {'before': '4778e1e0dedd0b1c9234b1c959b8f4084009fb93',
  'after': 'c124e62bc546fcfaebf0a442dfc50ea4befa5a7d'},
 'sample_type': 'before_fix',
 'trace': [{'idx': 0,
   'level': 0,
   'description': 'Unknown value from: apr_table_get',
   'func_removed': None,
   'file_removed': None,
   'file': 'modules/generators/mod_asis.c',
   'loc': '58:16',
   'func_name': 'asis_handler',
   'func_key': 'modules/generators/mod_asis.c@31:1-129:2',
   'is_func_definition': True,
   'url': 'https://github.com/apache/httpd/blob/4778e1e0dedd0b1c9234b1c959b8f4084009fb93/modules/generators/mod_asis.c/#L58'},
  {'idx': 1,
   'level': 0,
   'description': 'Assignment',
   'func_removed': None,
   'file_removed': None,
   'file': 'modules/generators/mod_asis.c',
   'loc': '58:5',
   'func_name': 'asis_handler',
   'func_key': 'modules/generators/mod_asis.c@31:1-129:2',
   'is_func_definition': True,
   'url': 'https://github.com/apache/httpd/blob/4778e1e0dedd0b1c9234b1c959b8f4084009fb93/modules/generators/mod_asis.c/#L58'},
  {'idx': 2,
```

Figure 1: An illustration of D2A dataset sample (i.e., first sample in HTTP dataset).

D2A contains samples from 6 well-known C++ open-source projects, as shown in Figure. 1. For the simplicity, in this project, we only select function field as our analysis feature, since we mainly focus on the natural language analysis, as show in Figure. 2. The research questions of this project includes:

1. Given a sample, does it contains some vulnerabilities?
2. How to represent the function syntax as numerical?
3. How to build a natural language model to identify bugs, and why does it work (explainability)?



d2a_lbv1_function_train

```
id  label  code
1       0  static void srt_to_ass(AVCodecContext *avctx, AVBPrint *dst,
                               const char *in, int x1, int y1, int x2, int y2)
           {
               if (x1 >= 0 && y1 >= 0) {
                   /* XXX: here we rescale coordinate assuming they are in DVD resolution
                    * (720x480) since we don't have anything better */

                   if (x2 >= 0 && y2 >= 0 && (x2 != x1 || y2 != y1) && x2 >= x1 && y2 >= y1) {
                       /* text rectangle defined, write the text at the center of the rectangle */
                       const int cx = x1 + (x2 - x1)/2;
                       const int cy = y1 + (y2 - y1)/2;
                       const int scaled_x = cx * (int64_t)ASS_DEFAULT_PLAYRESX / 720;
                       const int scaled_y = cy * (int64_t)ASS_DEFAULT_PLAYRESY / 480;
                       av_bprintf(dst, "{\\an5}{\\pos(%d,%d)}", scaled_x, scaled_y);
                   } else {
                       /* only the top left corner, assume the text starts in that corner */
                       const int scaled_x = x1 * (int64_t)ASS_DEFAULT_PLAYRESX / 720;
                       const int scaled_y = y1 * (int64_t)ASS_DEFAULT_PLAYRESY / 480;
                       av_bprintf(dst, "{\\an1}{\\pos(%d,%d)}", scaled_x, scaled_y);
                   }
               }

               ff_htmlmarkup_to_ass(avctx, dst, in);
           }
```

Figure 2: Various prediction tasks can be conducted on D2A, thefunction dataset that we are going to use.

3 Methods

Our methods inspired by the [1], and the workflow can be briefly divided by three parts: extracting functions, code snippet representation and deep learning vulnerability detection. Currently, we have obtained the code snippets and are working on building a abstract syntax tree for code representation. Then we are going to put forward a deep learning classification model to predict whether there are vulnerabilities in it. After all things are done, we will put the codes to GitHub as an open-source project. The demo will be in the form of Jupyter notebook, and we will dynamically show the features of D2A dataset, how the codes are represents, how the model are trained. The dataset is the D2A open-source dataset, which will not be put in our repository, but we will give a detailed instructions about how to use the dataset and our model.

References

[1] Uri Alon, Meital Zilberstein, Omer Levy, and Eran Yahav. Code2vec: Learning distributed representations of code. *Proc. ACM Program. Lang.*, 3(POPL):40:1–40:29, January 2019.

[2] Yunhui Zheng, Saurabh Pujar, Burn Lewis, Luca Buratti, Edward Epstein, Bo Yang, Jim Laredo, Alessandro Morari, and Zhong Su. D2a: A dataset built for ai-based vulnerability detection methods using differential analysis. New York, NY, USA, 2021. Association for Computing Machinery.